Improving Shipboard Applications of Non-Intrusive Load Monitoring

by

Richard A. Jones

M.S.E. Geodesy/Photogrammetry, Purdue University, 1997
B.S. Land Surveying Engineering, Purdue University, 1995
Submitted to the Department of Mechanical Engineering and
System Design and Management Program
in partial fulfillment of the requirements for the degrees of

Naval Engineer's Degree

and

Master's of Science in Engineering and Management

at the

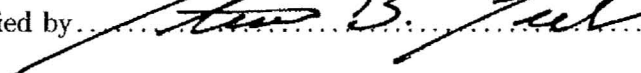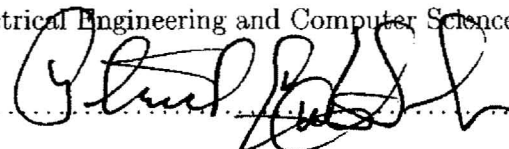MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author . . . .
Department of Mechanical Engineering and
System Design and Management Program
May 9, 2008

Certified by . . . .
Robert W. Cox
Assistant Professor of Electrical and Computer Engineering
University of North Carolina at Charlotte
Thesis Supervisor

Certified by . . . .
Steven B. Leeb
Professor of Electrical Engineering and Computer Science & Mechanical Engineering
Thesis Supervisor

Certified by . . . . . .
Patrick Hale
Director, Systems Design and Management Fellows Program
Thesis Supervisor

Accepted by . . . .
Lallit Anand
Professor
Chairman, Committee on Graduate Students

REPORT DOCUMENTATION PAGE

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
XX-06-2008	Master's Thesis	JAN-JUN 2008

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Improving Shipboard Applications of Non-Intrusive Load Monitoring	N62271-97-G-0026
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Richard A. Jones	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Massachusetts Institute of Technology	

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
Naval Postgraduate School Monterey, CA 93943	NPS
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

1. DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Non-Intrusive Load Monitor (NILM) measures equipment performance by measuring and analyzing the source power to the equipment at a single point in the electrical system. Previous studies have proven the usefulness of the NILM system in characterizing the state of mechanical systems onboard U.S. Coast Guard vessels and at the U.S. Navys Land Based Engineering Site (LBES) in Philadelphia, Pennsylvania.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	164	Sean Tibbitts, Educational Technician
					19b. TELEPHONE NUMBER (Include area code) (831) 656-2319 civins@nps.edu

Improving Shipboard Applications of Non-Intrusive Load Monitoring

by

Richard A. Jones

Submitted to the Department of Mechanical Engineering and
System Design and Management Program
on May 9, 2008, in partial fulfillment of the
requirements for the degrees of
Naval Engineer's Degree
and
Master's of Science in Engineering and Management

Abstract

The Non-Intrusive Load Monitor (NILM) measures equipment performance by measuring and analyzing the source power to the equipment at a single point in the electrical system. Previous studies have proven the usefulness of the NILM system in characterizing the state of mechanical systems onboard U.S. Coast Guard vessels and at the U.S. Navys Land Based Engineering Site (LBES) in Philadelphia, Pennsylvania.

This thesis seeks to augment the NILM system by exploring a more user friendly Graphical User Interface (GUI) to allow shipboard crews to utilize the NILM while in operation. Previous applications of NILM required post-event data analysis in the laboratory. An additional monitor was installed on the Low Pressure Air Compressor (LPAC) #1 at the LBES facility to investigate abnormalities detected in the operation of LPAC #2 by previous research. The ability of the NILM to function at the highest levels of the electrical distribution system was also explored at the LBES facility with the installation of two additional NILM systems on the main switchboards supplying power to the auxiliary system loads. Finally, a brief overview of the analysis software of the Multi-Function Monitor (MFM), a key component in modern ships Zonal Electrical Distribution Systems (ZEDS), is presented to explore the possibility of the NILM and MFM systems operating in conjunction to improve the operation of future ZEDS.

Thesis Supervisor: Robert W. Cox
Title: Assistant Professor of Electrical and Computer Engineering
University of North Carolina at Charlotte

Thesis Supervisor: Steven B. Leeb
Title: Professor of Electrical Engineering and Computer Science & Mechanical Engineering

Thesis Supervisor: Patrick Hale
Title: Director, Systems Design and Management Fellows Program

Acknowledgments

The author would like to acknowledge the following individuals for their assistance. Without them this thesis would not have been possible.

MIT Laboratory for Electromagnetic and Electronic Systems (LEES)

- Steve Leeb for allowing me to work on the NILM project.

- Rob Cox for assisting with many problems, any time of the day

- Jim Paris for handling the many computer related problems encountered

- Warit Wichakool for assisting in many ways and ensuring no one got hurt

- Chris Laughman for always having time for the odd question

Systems Design and Management

- Pat Hale for accepting me into the SDM program and providing valuable guidance along the way

- Bill Foley for assisting with the difficulties of being in the SDM and 2N programs

The Navy's Land Based Engineering Site (LBES)

- Charlie Gilligan for assisting with scheduling the many visits to the LBES facility

- Frank Facciolo for being patient and understanding to my many requests

- Lee Skarbek for providing technical assistance on any requested topic

- Andy Cairns for allowing me to utilize the LBES facility for my research

Family & Friends

- My wife, Diane, & children who endured abnormal work schedules and strange hours.

- Ayako Ito for invaluable assistance and knowledge

- My mother, Arline F. Jones (1936-2008) and my brother, Glen P. Jones (1959-2006) who saw me start on this endeavor but did not get to see me finish.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

13

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 NILM System

The Non-Intrusive Load Monitoring (NILM) system consists of two major components, an electrical enclosure that processes sensor inputs and a computer that analyizes the output of the electrical enclosure. On a 3-phase system, the inputs to the electrical enclosure are one current transducer (CT) and two voltages taps use to determine the voltage between two phases. The CT is normally installed on the A-phase and the voltage taps are installed on the B- and C-phases. The electronics in the electrical enclosure collects these measurements and sends them to the computer where the real and reactive powers are calculated. These values are sent to the computer which stores these values into hourly log files. These files can be analyzed to determine the operating characteristics of the shipboard systems and to perform diagnostics on these systems.

NILM research has be conducted at the Massachusetts Institute of Technologys (MIT) Laboratory for Electromagnetic and Electronic Systems (LEES) for over two decades. The main focus of the research for shipboard applications to this point has been to understand how the dynamics of various shipboard systems can be monitored using NILM and their operating modes and failure modes detected. The logging of the observed data occurred in real time as the events happened but the analysis was conducted at a later time.

1.2 Motivation for Research

1.2.1 NILM Graphical User Interface

The nature of the past research was focused primarily on understanding the systems being monitored. There is now a significant level of knowledge on a few specific systems such that the focus can be shifted to that of making the system useable and meaningful to the customer, in this case the ship's crew. In order to do this effectively, the analysis of the events that occurred and their classification had to be automated.

A second aspect that was explored was the interface through which the operator interacts with the NILM computer. Traditionally, the interface with the NILM system was a simple menu window with limited functionality. A significant number of operations required exiting to the Linux command line interface where standard Linux commands could be used. The processing of data involved utilizing scripts written in PERL and MATLAB^{TM}programs, both of which were normally written individually. It is unreasonable to assume that the ship's personnel for the U.S. Navy or the U.S. Coast Guard will have sufficient time or training to process the data in this fashion. Also, this would not lend itself to real time processing of the data and indication of the operation of the system being monitored.

1.2.2 NILM at the Land Based Engineering Site

The non-intrusive nature of the NILM system has had limited examination in a shipboard environment up to this time. In order to understand the operation of a particular system, the NILM equipment was normally installed electrically close to the system being monitored to minimize interference and noise in the signals. When installed in this manner, multiple system would be required to monitor many systems onboard the ship. This would not be truly non-intrusive as these systems would require many interfaces and NILM equipment located throughout the engineering spaces.

The Low Pressure Air Compressor (LPAC) monitored at LBES was found in previous work to have an unexplained abnormality in its operation [3]. In order to determine the cause of this abnormal signal in the operation of the LPAC, further investigations are required and the installation of a NILM on the second LPAC for comparison.

1.3 Objectives and Outline of Thesis

The research presented in thesis is a continuation of research conducted by LCDR William Greene, USN [18], LCDR Thomas D. McKay, Jr., USN [23], LCDR Patrick Bennett, USN [3] and LT Mark Piber, USCG [32]. LCDR Greene was on the team that installed the first set of two NILM sensors at the LBES Facility on the LPAC and the Fuel Oil (FO) Pumps supplying the LM2500 gas turbine main engines. He also presented the first look at using NILM in conjuction with the Multi-Function Monitors (MFMs) installed on ships equipped with a Zonal Electrical Distribution System (ZEDS). LCDR McKay installed a third NILM at the LBES facility on the Universal Engine Controller (UEC). Analysis of the data from the three NILM systems allowed major events which occur in the engineering plant. LCDR Bennett continued the study of the LBES facility, installing two additional NILM systems on the Lube Oil (LO) and Gas Turbine Generators (GTGs) to provide additional data for event change detection. LT Piber researched using NILM to diagnose and solve problems associated withthe Coast Guard Cutters.

The objective of this thesis is to further research into areas associated with the two main areas of the MIT LEES NILM group, the U.S. Coast Guard (USCG) Cutters and the LBES Facility. A Graphical User Interface (GUI) was investigated for use with the NILM system and was specifically designed for use with the CHT system onboard the USCG Cutters. At the LBES Facilty, research into the incorporation of the NILM system into the MFM systems. The MFM algorithm if examined in detail and specific functions related to NILM analysis. A second NILM was installed on the remaining LPAC at the LBES facility and the systems monitored to determine the cause of the abnormal operation noted be Bennett [3]. Two additional NILM systems were installed on the main electrical distribution panels for the LBES facility.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Graphical User Interface (GUI) For the NILM

Previous research with the NILM system focused on characterizing the operation of specific systems such that their operating states can be detected and used in diagnostics to determine normal and abnormal operating conditions. For example, detecting whether a pump is operating in slow speed or fast speed. As the systems were monitored by the NILM, the operating conditions that were logged by the operators were compared to the recorded data. This provides a means correlating the electrical power signals to known events. Once the events and their associated electrical signals have been correlated, the power signals can be monitored and the operating state of the equipment detected.

The installation of the NILM systems for the U.S.C.G. Cutters occurred while the ships were in port. Data collection and storage occurred while the ships operated at sea. Upon return to port, the data was downloaded and analyzed in a laboratory setting. The NILM system is effective in detecting abnormal modes of operation [32, 24] however the notification to the crew is always delayed by the need to analyze the data after returning to port. An automated classification algorithm was developed by LCDR Proper [34]. This thesis shows the development of a GUI for the Collection Hold and Transfer (CHT) system.

2.1 CHT System Description

The CHT system onboard the U.S.C.G. 270 foot Famous Class Cutters is a vacuum assisted collection system. The system consists of a 360- gallon collection tank, 2 sewage discharge

21

pumps, 2 vacuum pumps and the associated piping and valves necessary for controlling the flow. The system collects waste from 22 locations within the ship. The vacuum pumps maintain a vacuum within a normal operating range of 12 to 19 inches Hg. The sewage discharge pumps maintain the level within the collection tank as indicated by 3 level probes [17]. A simplified system diagram is shown in Figure 2-1.
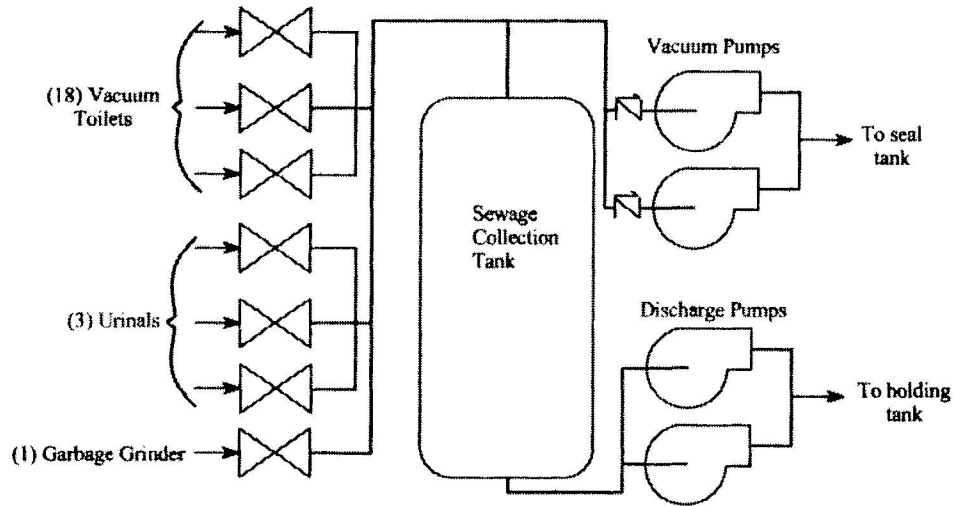
Figure 2-1: Basic Schematic of the CHT System [24]

2.2 CHT GUI Description

The CHT GUI was programmed in MATLAB™to take advantage of the advanced mathematics and GUI building features. The CHT GUI program was designed in a modular fashion with a systems perspective. The GUI would not be a stand alone program, it would take the output of the classification program developed by LCDR Proper [34] and present the information to the user.

2.2.1 Design

Requirements Identification

The requirements for the CHT GUI were to provide an interface for the ship's crew to utilize the NILM software without requiring extensive training. The design for the GUI needed to be intuitive such that crewmembers could easily understand the information quickly. The

22

NILM system would be intalled in an auxilary engineering space onboard the ship, not in an office. The crewmembers needed to be able to operate the GUI in this environment in the course of normal duties.

Design Philosophy

A modular design was developed with a clear definition of the interfaces between modules. Each module was designed to perform a specific function and provide the resultant data or indications to the other modules through the interface variables. The GUI was designed specifically for the CHT system onboard the USCG cutters, but with a modular design, modules could be changed to make it applicable to other systems. The modular system would also facilitate upgrades to the system as better processing algorithms are developed. A symplified design spiral approach was used in the development of the GUI.

2.2.2   Implementation of GUI

The design of the GUI was performed with utilizing concepts of human interface design. Large buttons were included that could be utilized via the touch screen and would not require fine motor controls such as drop down menus do. Since the GUI would be located in an engineering space, fine controls would make the interface more difficult for the crew to use.

The GUI has multiple screens varying in the depth of the information presented. The intitial screen contains a summary of the system operation and have all of the information needed during normal operation and is shown in Figure 2-4. A second window provides more detailed information in the event a fault is detected. The summary screen contains a simplified diagram of the system which is updated by the program to reflect the status of the system. The color of the equipment is changed as the status changes the color scheme is intuitive and shown in Table 2.1.

Table 2.1: Color Scheme for System Diagram

None	Equipment secured (pumps only)
Green	Normal Operation
Yellow	Potential Fault Detected
Red	Fault Detected
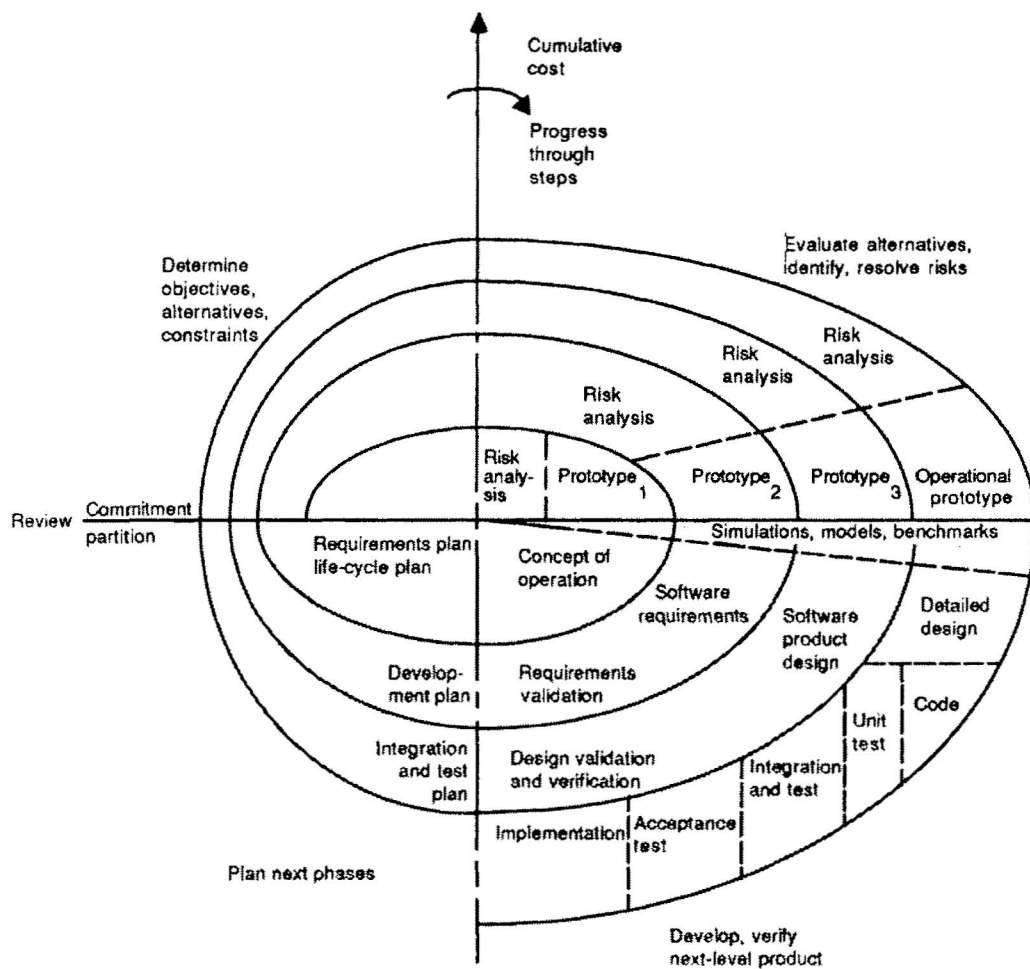
Figure 2-2: Spiral Development Cycle of Software[4]

When the system is operating normally, the equipment color will be changed to green to indicate a component is operating. An example of this is shown in Figure 2-5 which indicates that a Vacuum Pump has started and is opertating normally. Yellow is displayed when a fault is detected but for an insificient number of samples to flag it as a fault. This threshold setting prevents spurious alarms in the system but allows the abnormal indications to be logged and categorized for further analysis.

When a fault in the system is received, an example of which is shown in Figure 2-6 as a Level Probe Failure, the component color is changed to red and a audible warning is given. To access the detailed information screen, the operator clicks on the component using the pointing device or the electronic pen included with the tablet PC. An example of

24
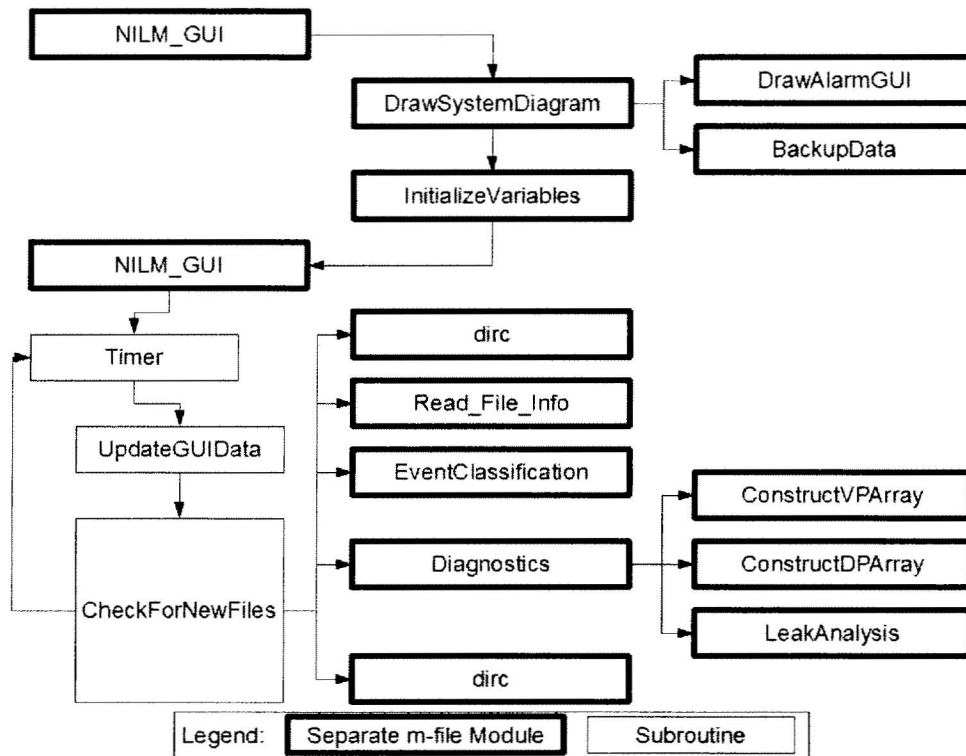
Figure 2-3: Data Flow Diagram of CHT GUI Program.

the detailed information screen is shown in Figure 2-7.

The failures indicated by the system were those identified through previous research on the CHT system [24, 32] and are listed in Table 2.2. The Vacuum Pump Seal Water Line Clog failure indication is provided by the classification program [34]. The remaining failures are detected by routines within the GUI program. The failure detection algorythms are contained within separate modules of the program. This allows modifications to be made quickly and easily if the algorithms are improved or require adaptation if the system is installed on a different class of ship. As an indication of the effectiveness of this approach, the leak detection algorithms were developed by previous research [24, 32] and are utilized by this program by a standard data exchange interface.
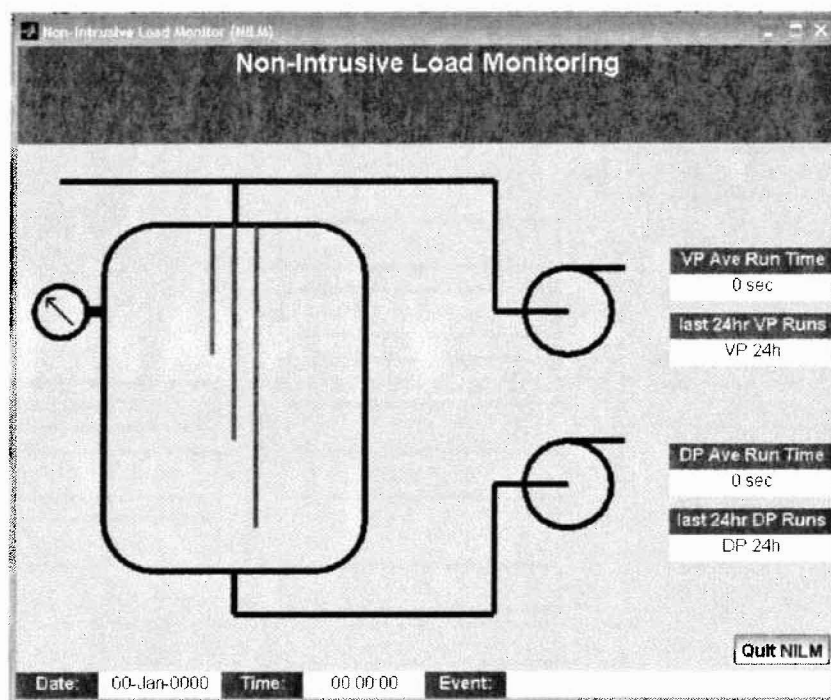
Figure 2-4: CHT GUI at Startup. Shown in the diagram are the Collection Tank, the Vacuum Gauge, one Vacuum Pump representing both physical pumps and one Discharge Pump representing both physical pumps.

Table 2.2: Failures in the CHT System Indicated on the GUI Interface

Vacuum Tank Level Probe Failure
Vacuum Pump Seal Water Line Clog
Vacuum Line Clog
Small System Leak
Large System Leak
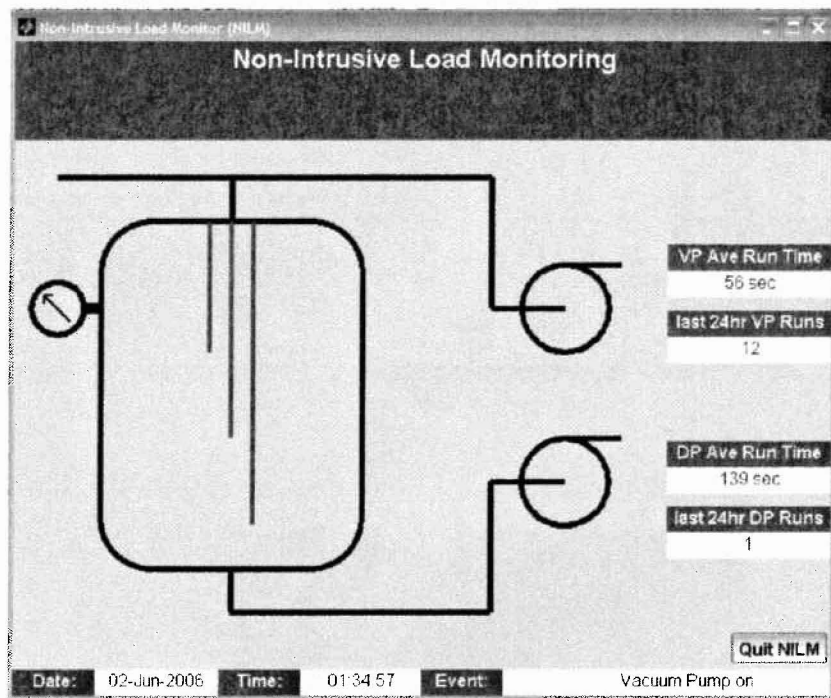
Figure 2-5: CHT GUI with Vacuum Pump On
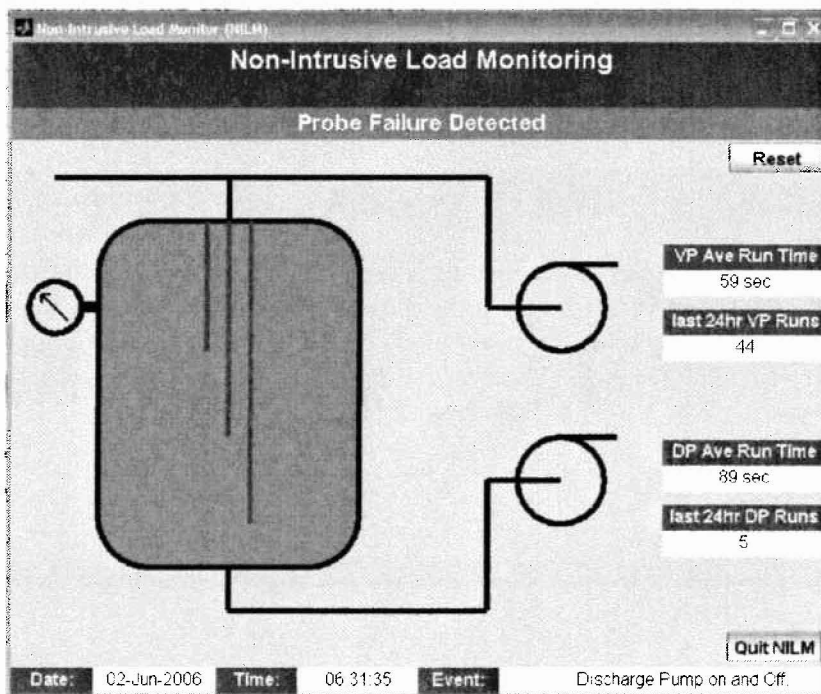
Figure 2-6: CHT GUI with Indication of Level Probe Failure
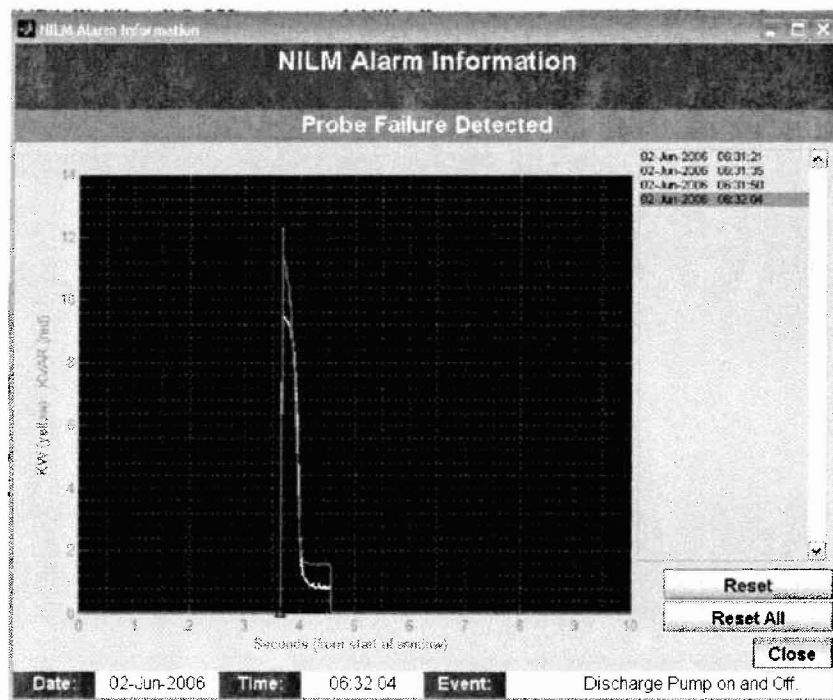
Figure 2-7: CHT GUI Alarm Information Screen. The yellow trace represents the Real Power in KW and the red trace the Reactive Power in KVAR. This screen shows the power traces from a Discharge Pump start and stop. The Discharge Pump normal run time is approximately 45 to 60 seconds. These traces show a run time of approximately 1 second and is an indication of a Vacuum Tank Level Probe failure.

Chapter 3

Land-Based Engineering Site (LBES)

3.1   Background

The LBES facility provides an opportunity to test the functionality and operation of the NILM equipment on Navy systems without impacting the operation of Navy ships. The LBES facility is located on the Naval Business Center in Philadelphia, Pennsylvania. The LBES was built to replicate the equipment and systems of the #2 Engine Room of the Navy's Arleigh Burke (DDG-51) class of destroyers. The facility provides for testing by the U.S. Navy and for contractors working on shipboard systems. It is also utilized by the crews of newly built destroyers as a means of training prior to operating their engineering equipment for the first time.

The equipment installed at LBES are the major systems found in #2 Engine Room: two LM2500 gas-turbines propulsion main engines (GTMs) along with the reduction gears, shafting and bearings, three gas-turbine generators (GTGs), and their associated auxiliary systems such as Fuel Oil (FO), Lubricating Oil (LO), Low Pressure Air (LP Air) and cooling water.

Since it is a land-based test facility, LBES is configured differently than the #2 Engine Room on a DDG-51. For instance, LBES has a complete Zonal Electrical Distribution System (ZEDS) with all three GTGs instead of the single GTG installed in #2 Engine Room on DDG-51 class ships. Since the LBES is a land-based facility, a large waterbrake

is installed to simulate the Controllable Pitch Propeller (CPP) that would normally be on the end of the propulsion shaft. The shaft length at LBES is shorter than it is on a DDG-51 class ship, but the waterbrake appears the same as a CPP to the propulsion train [23, 3]. A perspective view of the LBES facility is shown in Figure 3-1 and a picture is shown in Figure 3-2
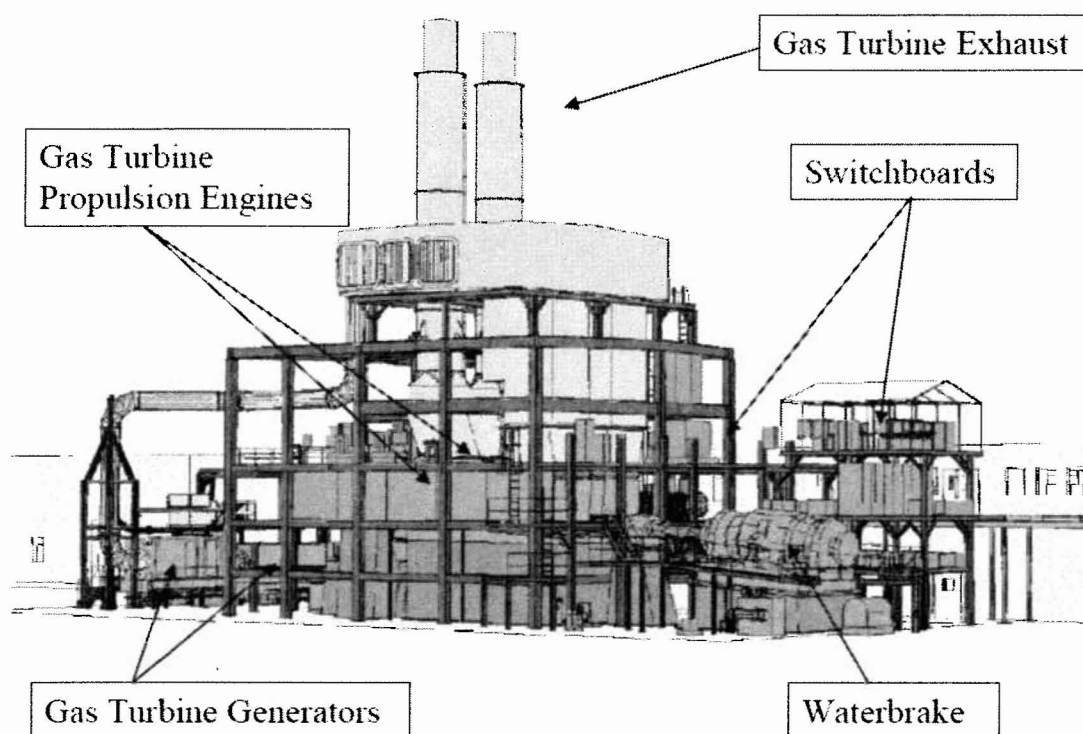
Figure 3-1: Layout of LBES Facility in Philadelphia, PA [18]

3.2 Monitored Systems

The following is a description of the NILM systems installed to monitor the equipment at the LBES facility. Each section has a short summary of the equipment monitored and the NILM installation parameters needed to process the data. A diagram of a portion of the electrical distribution system used to power engineering loads is shown in Figure 3-3. The location of the NILM sensors is indicated in the figure by the red rectangles. The LBES facility has two electrically isolated distribution systems. The engineering space equipment is powered from commercial sources. The output of the electrical generators in

30
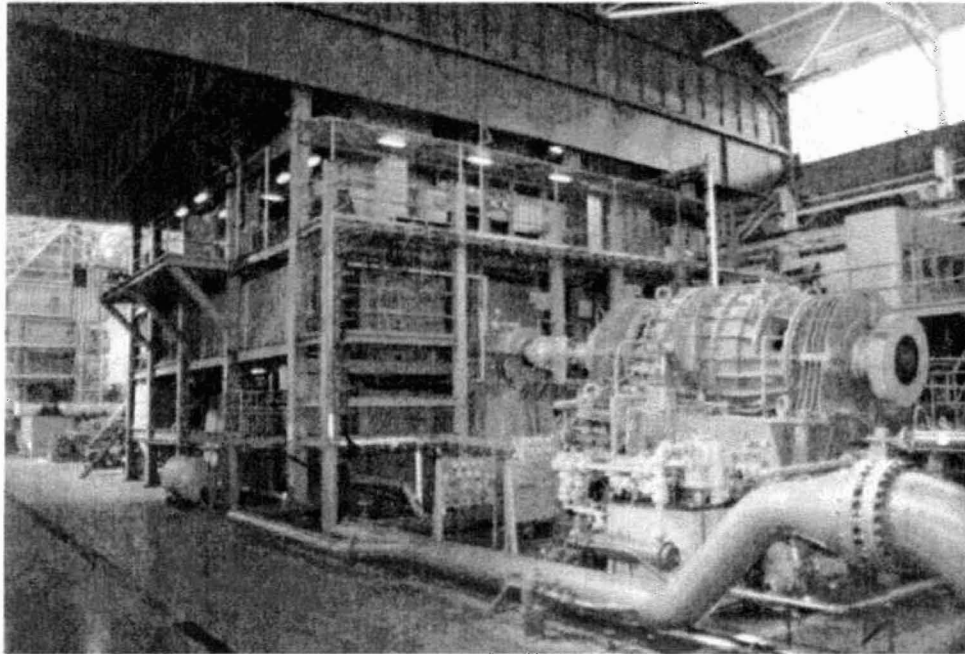
Figure 3-2: Picture of LBES Facility in Philadelphia, PA [23]. The waterbrake is in the foreground. The propeller shaft can be seen leading from the waterbrake to the shaft bearing, which is located just aft of the LM2500 Gas Turbines.

the engineering spaces is connected to a bank of load cells. Therefore no NILM systems are installed on that portion of the electrical distribution system.

3.2.1 Low Pressure Air Compressor #2

The LPAC #2 NILM system was installed by LCDR Greene [18]. The NILM for LPAC #2 is colocated with the NILM for LPAC #3 because the LPACs are physically located adjacent to each other. Each LPAC provides pressurized air for the use in pneumatic valve control and pneumatic power for various equipment. Only one LPAC is required to supply all loads at the LBES facility. In the event both LPACs are not operational, the LP Air system is supplied via a site air system. It should be noted that there is no LPAC #1 at LBES because the on a DDG-51 LPAC #1 is located in #1 Engine Room.

The site LP Air system consists of two large, high capacity air compressors, air receivers and associated valves and piping which supply compressed air to the building in which the LBES facility is located. An isolation valve (FA-V007) and check valve (APL-V003) are located in the line where the site LP Air system connects to the LBES LP Air system. The

31
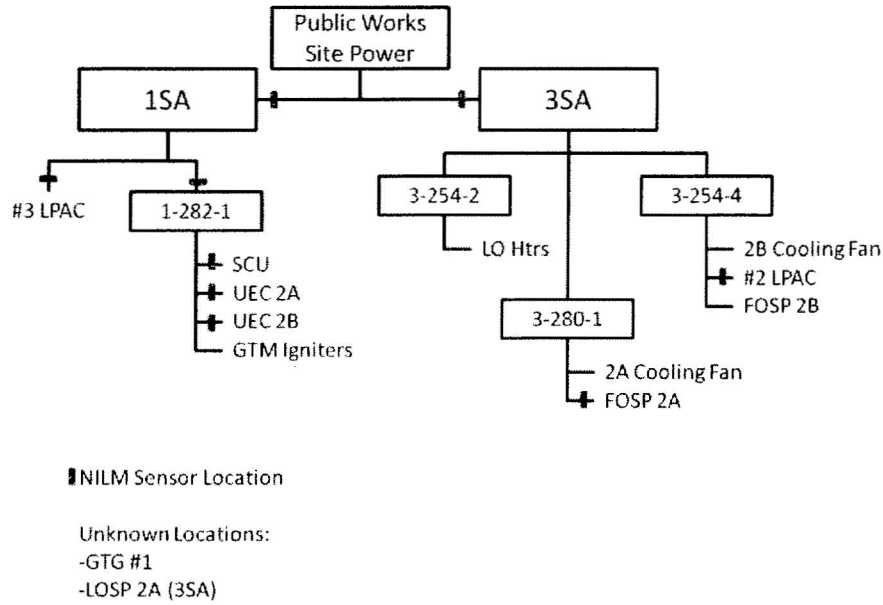
Figure 3-3: LBES Electrical Distribution Diagram shown the relevant loads that are monitored by NILM systems.

check valve prevents the LBES LP Air system from supplying air to the site system.

The NILM voltage tap and CT sensor are installed inside the controller for the LPAC. The details of the NILM configuration are shown in Table 3.1. The physical installation is shown in Figure 3-5 and Figure 3-4.

Table 3.1: LPAC #2 NILM Data

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	A and B	180Ω	LEM LV-25P	0
2	LPAC Current	C	36Ω	LEM LA-305S	0
3	Pressure	None	None	Omega PX209	0
The LPAC #2 NILM is set to monitor 4 channels of data in raw data mode.					
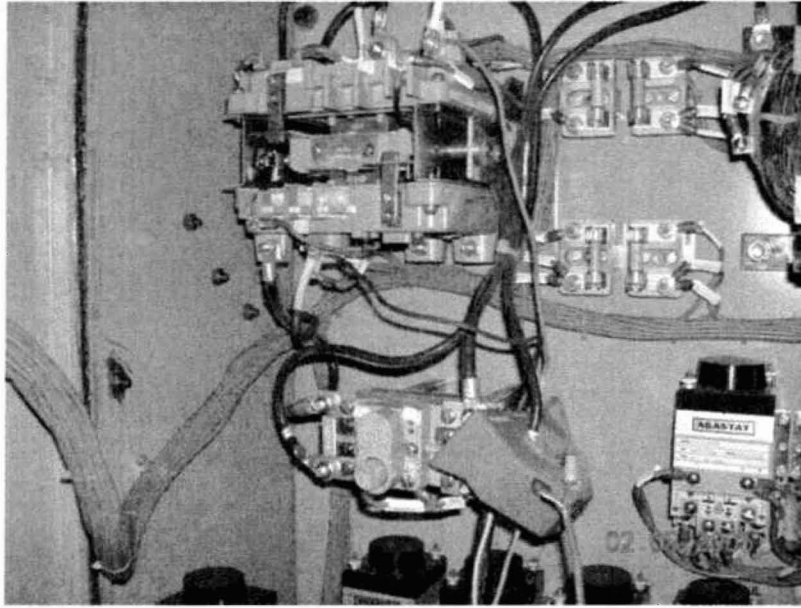
Figure 3-4: NILM Intallation on LPAC #2. The CT is connected on phase A and the voltage tap is connected across phases B & C.

3.2.2 Low Pressure Air Compressor #3

The Low Pressure Air Compressor (LPAC) #3 NILM was installed as part of the research for this thesis. The NILM voltage tap and CT sensor are installed inside the controller for the LPAC. The details of the NILM configuration are shown in Table 3.2. The physical installation is shown in Figure 3-5 and Figure 3-6.

Table 3.2: LPAC #3 NILM Data

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	B and C	130Ω	LEM LV-25P	0
2	Current	A	13.5Ω	LEM LA-305S	0
The LPAC #3 NILM is set to monitor 2 channels of data in raw data mode.					

3.2.3 Gas Turbine Generator Controller

Gas Turbine Generator (GTG) #1 was chosen for monitoring because of it is equiped with a Redundant Independent Mechanical Start System (RIMSS). The RIMSS consists of a small DC motor, a small gas producing turbine and a small RIMSS power turbine connected to

33
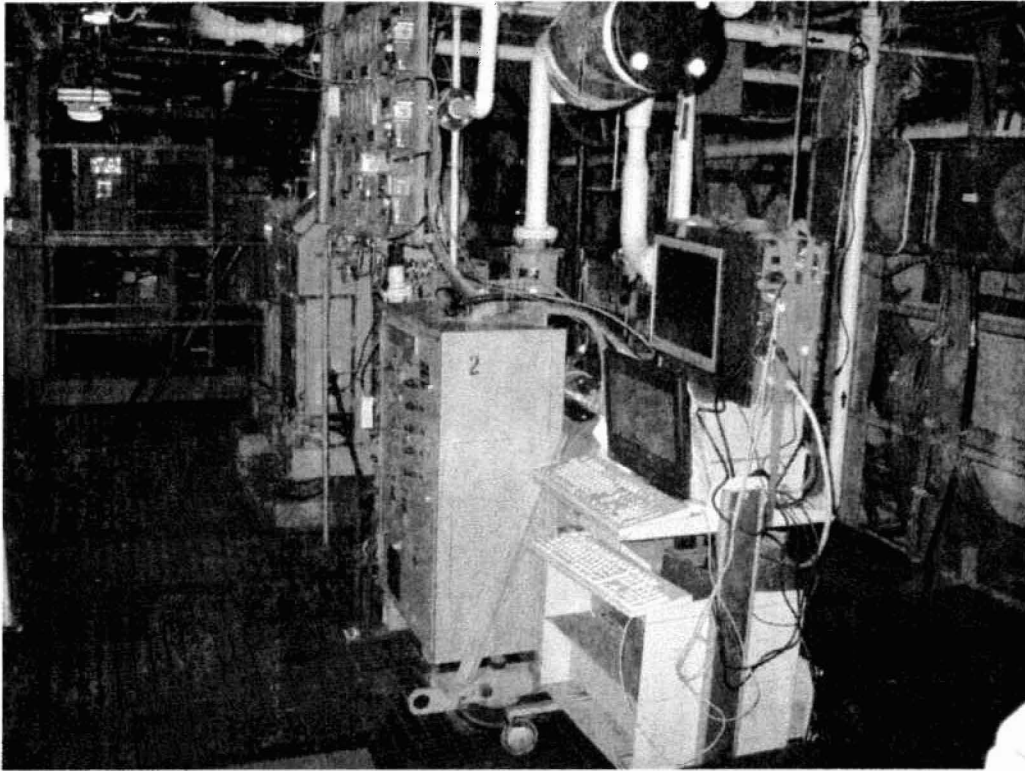
Figure 3-5: LPACs #2 and #3 at LBES. Both NILM Monitoring systems are installed on the cart in the foreground of the picture.

Voltage Tap
Phases B & C

Current Transformer
Phase A
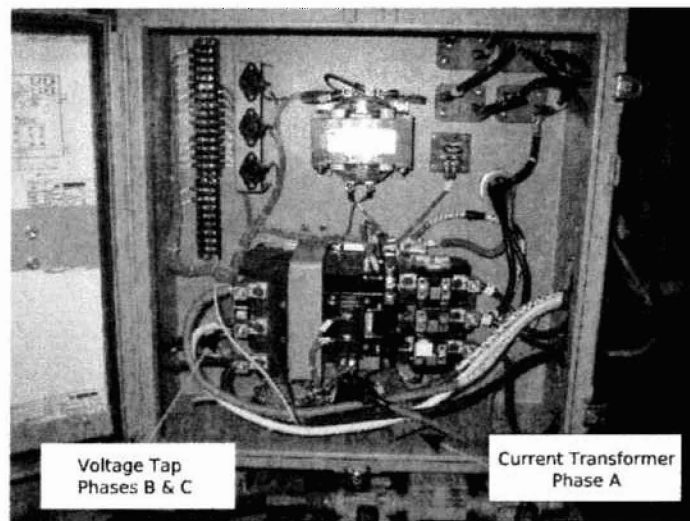
Figure 3-6: NILM Intallation on LPAC #3. The CT is connected on phase A and the voltage tap is connected across phases B & C.

the reduction gear of the GTG. The GTG is designated AG9140RF, which indicates that it is an Allison 501-K34 gas turbine and a KATO 3000 kW electrical generator. GTG #1 can operate independent of the ship's electrical distribution system to allow it to be started in completely dark conditions. To accomplish this, it is equiped with a No Break Power Supply (NBPS) which is essentially an uninteruptable power supply for the Full Authority Digital Control (FADC) Local Operating Panel (LOCOP). The FADC LOCOP controls and monitors the gas turbine, the RIMSS, the electrical generator and all associated auxiliary systems. The NILM is connected to the FADC LOCOP to monitor all control signals to the GTG.

Table 3.3: Gas Turbine Generator NILM Data

NILM Channel	Measurement	Phase	Measurement Resistors	Reference Resistors	Tranducers	Gain Code
1	Voltage	A and B	110Ω	51Ω	LEM LV-25P	0
2	Current	C	147Ω	51Ω	LEM LA-305S	0
The GTG NILM is set to monitor 2 channels of data in preprocess mode.						
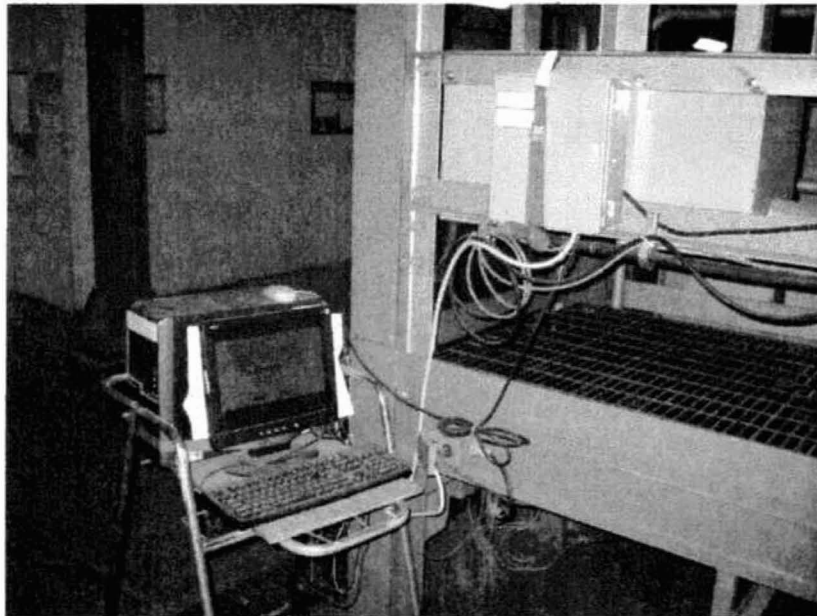
Figure 3-7: NILM Intallation on the GTG

3.2.4 Gas Turbine Propulsion Main Engine Controller

The Gas Turbine Propulsion Main Engine (GTM) Controller NILM monitors Power Panel 1-282-1 and three specific loads on the panel: the Universal Engine Controller (UEC) Plus for the 2A engine, the UEC Plus for the 2B engine, and the the Shaft Control Unit (SCU). Power Panel 1-282-1 provides 120VAC power to these components and to the GTM igniters and the GTM module light. The NILM system in this location monitors all four CTs and the voltage tap on the power panel which is common to all channels. Figure 3-8 shows the power panel and Figure 3-9 shows the CTs within the power panel. The details of the NILM configuration are shown in Table 3.4, and details on each monitored system (i.e. UEC, SCU) are presented in separate subsections below.
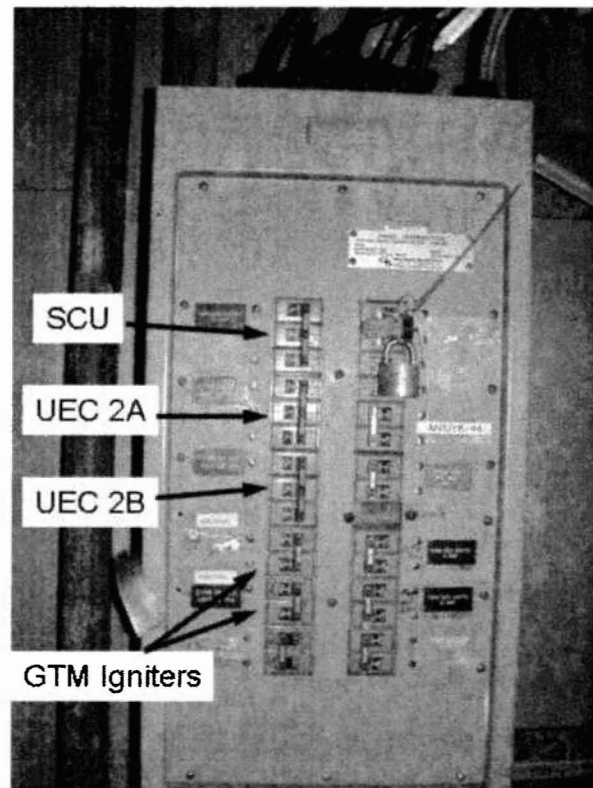
Figure 3-8: NILM Intallation on Panel 1-282-1 [3]. Several loads on the panel are annotated. The SCU, UEC 2A and UEC 2B are separately monitored.
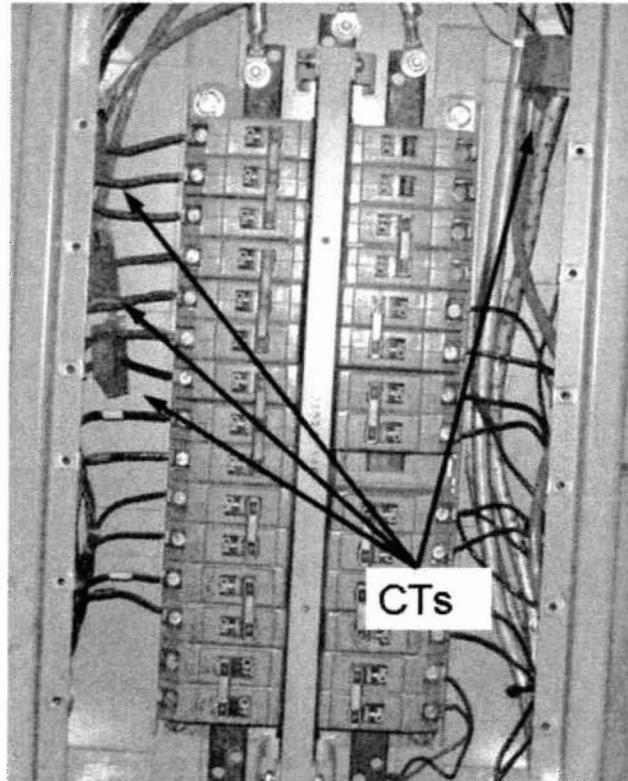
Figure 3-9: NILM CT Connections on Panel 1-282-1 [3]. The CTs for the individual equipment are on the left of the panel and the aggregate panel CT is located in the upper right.

Table 3.4: Panel Aggregate Current NILM Data [3]

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	B and C	110 Ω	LEM LV-25P	0
2	UEC 2A Current	A	130Ω	LEM LA-55S	3
3	UEC 2B Current	A	130Ω	LEM LA-55S	3
4	SCU Current	A	130Ω	LEM LA-55S	2
5	Panel Current	A	67Ω	LEM LA-205S	1
The GTM NILM is set to monitor 6 channels of data in raw data mode.					

Universal Engine Controller Plus 2A & 2B

The Universal Engine Controller (UEC) Pluses provide all control and monitoring functions for the LM2500 gas turbines [29]. Each GTM has a separate UEC that operates

independently of the other. The control signals required to start and operate the GTM can be observed by monitoring the power to the UEC. The necessary monitoring and analysis process is described in [3]. Detailed information on the UECs can be found in [31].

Shaft Control Unit

The Shaft Control Unit (SCU) provides a local control interface to both GTMs. The control functions include starting and stopping the gas turbine, fuel valve control, Lube Oil Service pump control, and Fuel Oil Service Pump Control. The SCU is normally unmanned with control of the GTMs at the Propulsion and Auxiliary Control Console (PACC) located in the LBES Central Control Station (CCS). The PACC is connected to the SCU via fiber optic cabling [3]. The SCU is monitored by the NILM to enable the observation of the control signals for the control of the GTMs and their associated auxiliary equipment. The necessary monitoring and analysis process is described in [3]. Detailed information about the SCU can be found in [28].

3.2.5   Fuel Oil Service Pump #2A

The Fuel Oil Service Pumps (FOSP) supply fuel to the GTMs directly and to the gravity head tank used by the GTGs. The FOSPs are two-speed pumps. Only one pump is required for normal operation of the engineroom equipment. 2A FOSP is monitored by a NILM system. The details of the NILM configuration are shown in Table 3.5, and a photo of the NILM installation is shown in Figure3-10 and Figure 3-11.

Table 3.5: Fuel Oil #2A NILM Data

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	A and B	177Ω	LEM LV-25P	0
2	2A FOSP Current	C	61Ω	LEM LA-55S	0
The FO NILM is set to monitor 2 channels of data in preprocess mode.					
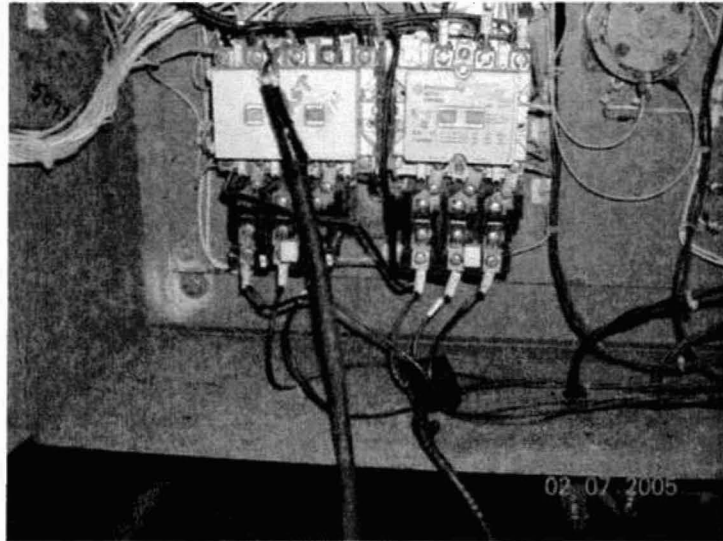
Figure 3-10: NILM Intallation on Fuel Oil Service Pump #2A [18].
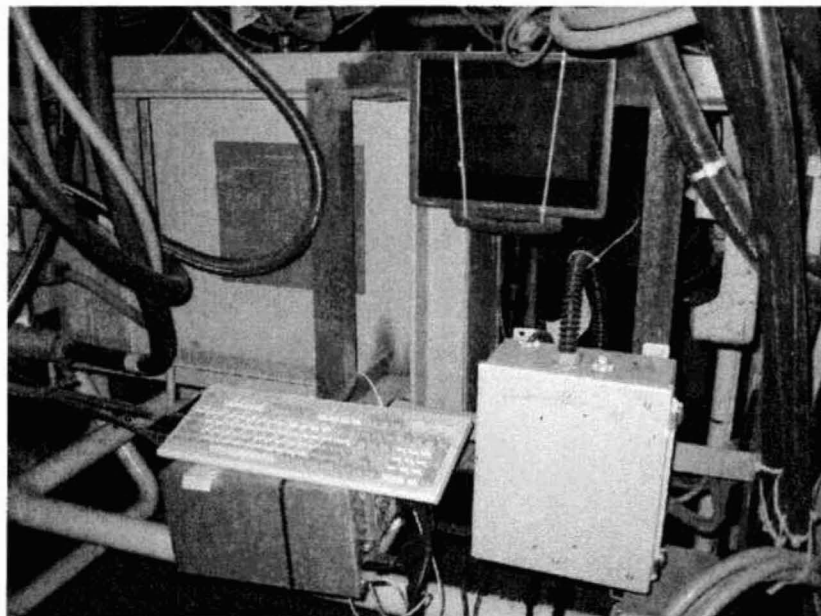
Figure 3-11: NILM Intallation on the 2A FO Pump

3.2.6 Lube Oil Service Pump #2A

The Lubricating Oil Service Pump (LOSP) supplies lubricating oil to the Main Reduction Gear (MRG) and the propulsion shaft thrust bearing. The LOSP has two operating speeds. 2A LOSP is monitored by a separate NILM. The details of the NILM configuration are

shown in Table 3.6 and a photo of the NILM installation is show in Figure 3-12 and Figure 3-13.
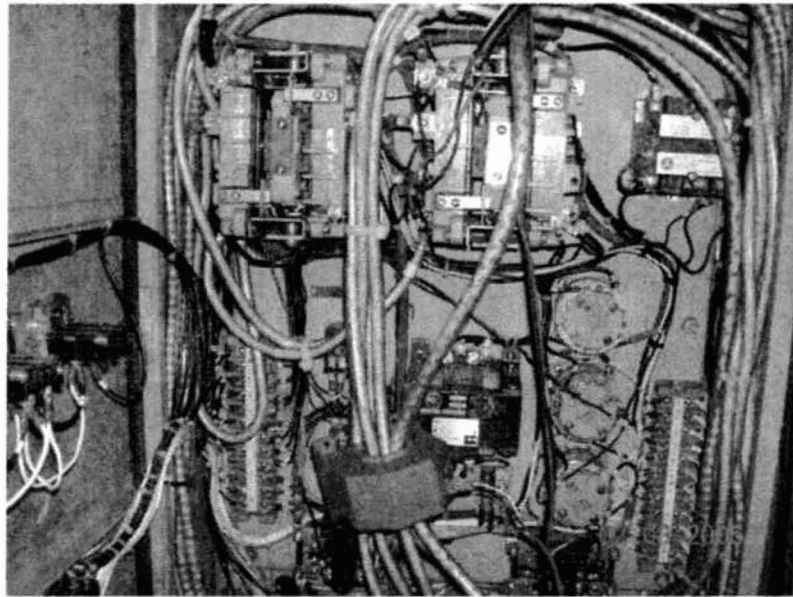
Figure 3-12: NILM Intallation on Lube Oil Pump #2A. [18]

Table 3.6: Lube Oil Pump #2A NILM Data

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	A and B	110Ω	LEM LV-25P	0
2	LOSP Current	C	20Ω	LEM LA-305S	0
The LO NILM is set to monitor 2 channels of data in preprocess mode.					

3.2.7   1SA & 3SA Switchboards

The 1SA and 3SA switchboards are located on the upper level of the LBES facility. They are supplied by a public utility through the circuit breakers "MAIN BKR. 1SA-M1" and "MAIN BKR. 3SA-M1". There are two NILMs installed, one on each switchboard, with their sensors connected on the upstream side of these circuit breakers. The voltage taps are on phases A & B and the CTs are on phase C. The connections to panel 3SA are shown in Figure 3-14. The connections to 1SA are similar.

Figure 3-13: NILM Intallation on the 2A LO Pump

The 1SA and 3SA NILMs each record four channels: two for voltage and two for current. The CTs used in these installations are different than the ones normally used in NILM installations. The primary reason for this change is that the test director requested the use of a split-core CT in order to avoid breaking electrical connections. A Fluke i3000S AC Clamps were chosen for use as the CTs. The specification sheets for the Fluke clamps are presented in Appendix A.

Both switchboard NILMs use Fluke i3000's CTs set to the 3000A range. At this setting, they are not damaged by the typical operating currents, but they do produce very small output voltage signals. An amplifer circuit was designed by Professor Cox to raise the signal to a more reasonable level. The amplifer card has two amplifiers installed, and the single input from the CT is split and amplified at two levels in the current configuration, one at 10x and one at 100x. Further testing will determine the necessary amplification levels needed. The levels can be easily adjusted by changing resistors in the circuit. The amplifier card is shown in Figure 3-15.

The details of the NILM configuration are shown in Table 3.7 and Table 3.8.

Table 3.7: 1SA NILM Configuration

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	A and B	110Ω	LEM LV-25P	1
2	Current	C	None	Fluke i3000S	1
3	Voltage	A and B	110Ω	LEM LV-25P	1
4	Current	C	None	Fluke i3000S	4
The 1SA NILM is set to monitor 4 channels of data in preprocess mode.					

Table 3.8: 3SA NILM Configuration

NILM Channel	Measurement	Phase	Measurement Resistors	Tranducers	Gain Code
1	Voltage	A and B	110Ω	LEM LV-25P	4
2	Current	C	None	Fluke i3000S	1
3	Voltage	A and B	110Ω	LEM LV-25P	4
4	Current	C	None	Fluke i3000S	4
The 3SA NILM is set to monitor 4 channels of data in preprocess mode.					

3.3 Data Analysis

During the course of this thesis project, two tests were performed on the LBES systems. The first test was to simultaneously record power signals at both the auxiliary equipment and at the switchboards. The second was to determine the cause of abnormal signals detected in the LP Air system by previous researchers [3, 23].

3.3.1 Power Signal Test of Auxiliary Equipment

The NILM systems from all previous research were installed locally at each load or on a power panel near the circuit breaker for the load. To test the effectiveness of installing NILMs at the main distribution switchboards each load was started and operated in all available modes while the local NILM and the main switchboard NILMs were monitoring the system. Neither the GTMs nor the GTGs were available for operation during the testing period. Therefore, for the test of the 3SA NILM, LPAC #2 was started first in automatic mode to provide some variation in the signal as the LPAC cycled during it's normal operation. The other loads known to be powered from this switchboard were the 2A LOSP and 2A FOSP. The LOSP is the larger of the two motors and was chosen to be
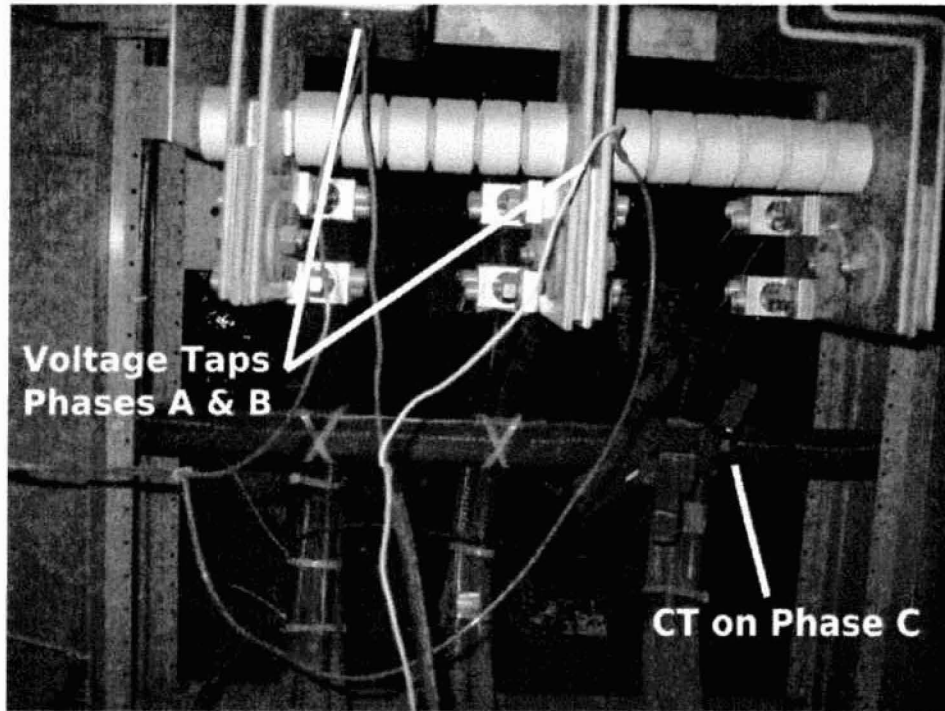
Figure 3-14: NILM Intallation on the 3SA Switchboard. The connections for the 1SA Swithboard are similar. The voltage tap wires can be seen on Phases A & B. The Fluke CT can be seen clamped around Phase C.
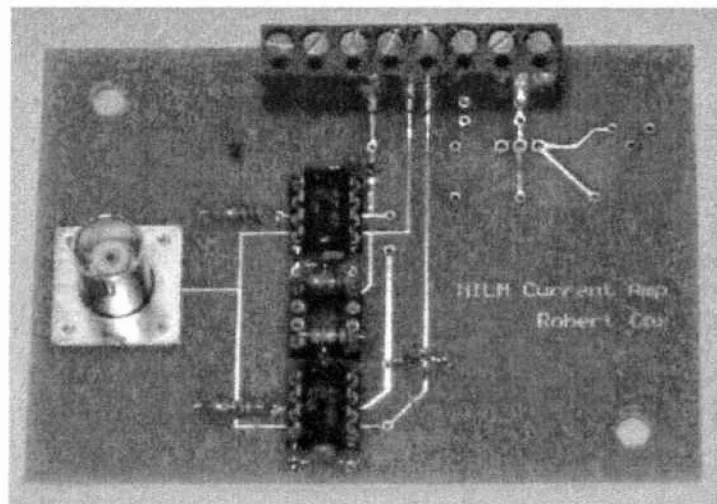
Figure 3-15: Amplifier card used to amplify the signal from the Fluke CT for use within the NILM.

Figure 3-16: Photograph of the NILM location for the 1SA & 3SA Switchboards.

started first. It was started in slow speed, ran for approximately one minute then shifted to fast speed without stopping. The FOSP was then started in slow speed and shifted to fast speed approximately one minute later. It was opperated at fast speed then shifted back to slow speed prior to securing. Once the FOSP was secured, the LOSP was secured. Figure 3-17 shows the results of this test. Table 3.9 gives an explanation of the events annotated in the figure. The results of the test provided the expected indications and all signals were detectable on both the local NILMs and on the 3SA NILM.

The second portion of the test was to perform a similar test for the loads powered from the 1SA switchboard. LPAC #3 was started first and placed in automatic mode to provide a cycling load while the loads on the switchboard were operated. The loads thought to be powered from 1SA were the 'B' series of pumps, 2B LOSP and 2B FOSP. These pumps do not have local NILMs installed to monitor them, therefore, only the 1SA NILM was expected to record their operating signals. The 3SA NILM was also operating during this test but was not expected to gather any useful data. The LOSP and the FOSP were operated in the same order and manner as during the first test. It was noted while processing the data that the 'B' series of pumps are indeed powered off of the 3SA switchboard. The only load monitored on the 1SA switchboard during the test was LPAC #3. The LOSP and the FOSP

44

were monitored by the 3SA NILM and as shown in Figure 3-18 were operated without any cycling load. Table 3.9 gives an explanation of the events annotated in Figure 3-18.

The NILM monitors installed on the 1SA and 3SA switchboards are able to monitor other equipment that has not been monitored previously at the LBES facility. An example of the power trace from one piece of equipment is the FO Purifier shown in Figure 3-19. Table 3.11 gives an explanation of the events annotated in the figure. The interesting shape of the power curve is due to the eight-step starting sequence of the FO Purifier, which is necessary due to the rotating mass of the *bowl*. The *bowl* is the internal assembly that rotates at a high rate of speed and is used to remove impurties from the FO. An interesting event occurs in the shutdown sequence that is detectable in the power trace. The operator "shoots the bowl" with water to provide some cleansing of the bowl during shutdown[1]. This step is peformed twice per procedure. This process can be detected by the NILM and is shown in Figure 3-19.

Another aspect of observing the aggregate signal is that the *signals of interest* will be contained in background *noise* of other loads at LBES. An example of this is shown in Figure 3-20 which shows the effects of the Lube Oil (LO) Heaters operating and LPAC #2 is cycling simultaneously. Table 3.12 gives an explanation of the events annotated in the figure. The LO Heaters have this type of signal because of the power modulation of the signal to give a relatively high average power output of the heaters without damaging the heaters due to excessive current. The cycling frequency of the signal is shown in Figure 3-21. The annotation in the figure is the same as in Figure 3-20. The cycling of LPAC #2 can be seen in the overall signal at the lower edges of the power levels of the cycling heaters. This can be effective when observing only two loads, however, when multiple loads are operating the individual signals will not be able to be extracted easily. A potential solution for this problem is the *ginzu* program detailed in [34].

---

[1]The theory of operation of the FO Purifier was obtained through discussion with Frank Facciolo at the LBES facility.
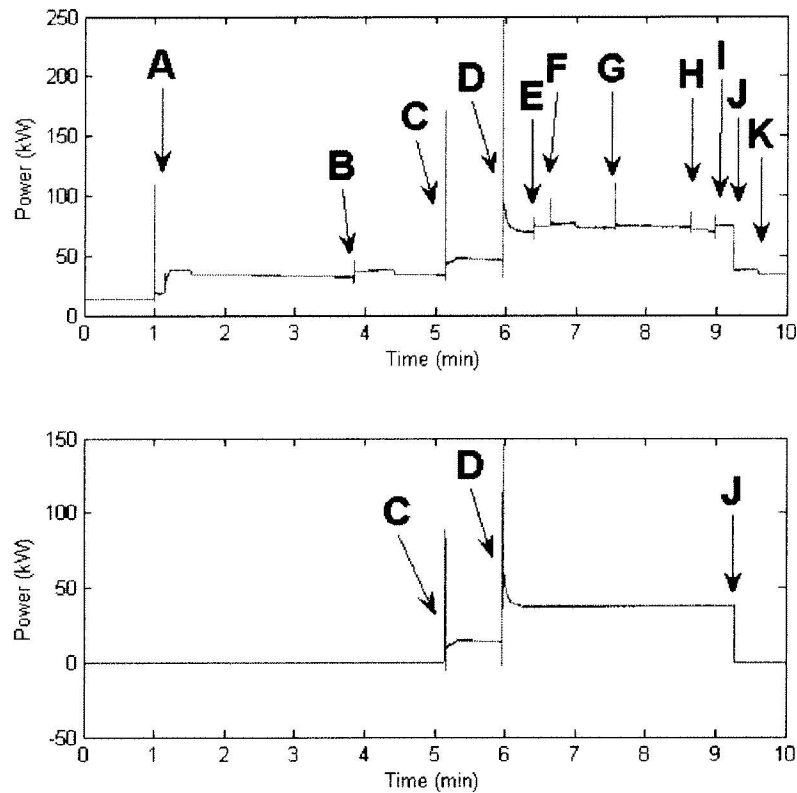
Figure 3-17: Plot of Power on 3SA Switchboard during Power Test Procedure. The events are annotated with letters and are explained in Table 3.9

Table 3.9: Explanation of events in Figure 3-17

Event	Description
A	LPAC #2 Started in Automatic
B	LPAC #2 cycling
C	2A LOSP Started in Slow Speed
D	2A LOSP Shifted to Fast Speed
E	2A FOSP Started in Slow Speed
F	LPAC #2 cycling
G	2A FOSP Shifted to Fast Speed
H	2A FOSP Shifted to Slow Speed
I	2A FOSP Secured & LPAC #2 cycling
J	2A LOSP Secured
K	LPAC #2 cycling
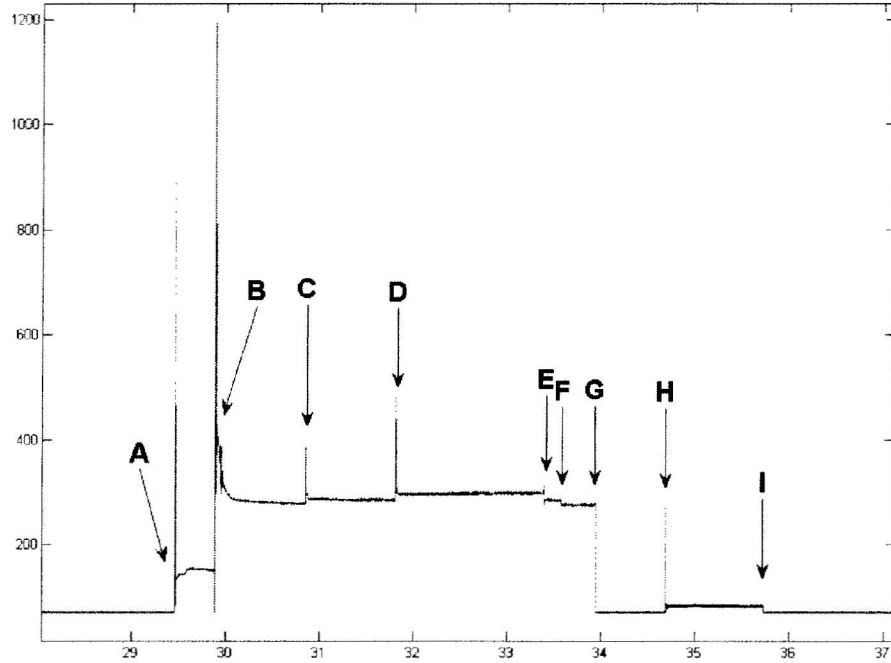
Figure 3-18: Plot of Power on 3SA Switchboard during Power Test Procedure Showing 'B' Pumps. The events are annotated with letters and are explained in Table 3.10

Table 3.10: Explanation of events in Figure 3-18

Event	Description
A	2B LOSP Started in Slow Speed
B	2B LOSP Shifted to Fast Speed
C	2B FOSP Started in Slow Speed
D	2B FOSP Shifted to Fast Speed
E	2B FOSP Shifted to Slow Speed
F	2B FOSP Secured
G	2B LOSP Secured
H	Waste Pump Started
I	Waste Pump Secured

47
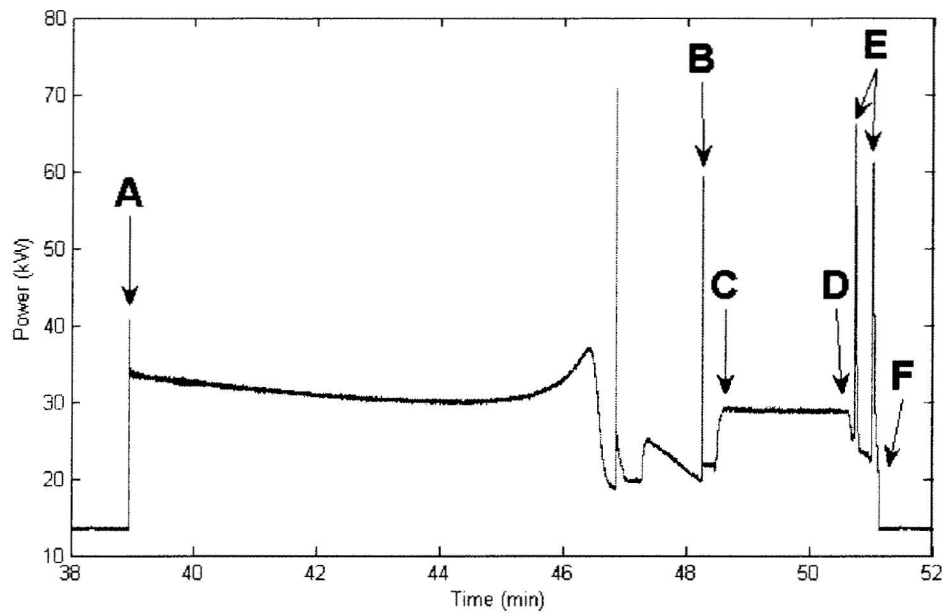
Figure 3-19: Plot of Power for the Start of the Fuel Oil Purifier. The events are annotated with letters and are explained in Table 3.11

.

Table 3.11: Events for the Start of the Fuel Oil Purifier Corresponding to Figure 3-19

Event	Description
A	FO Purifier Start Sequence Commenced
B	FO Start Sequence Complete, FO Purifier FO Pump Started
C	FO Purifier Online
D	FO Purifier Offline
E	"Shooting the Bowl"
F	FO Purifier Secured

48
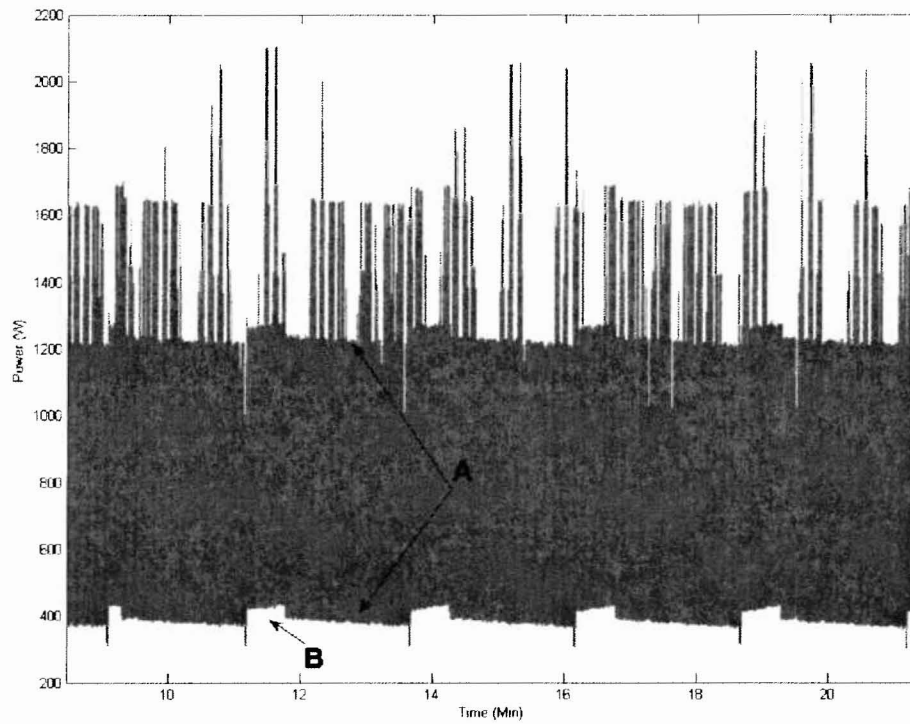
Figure 3-20: Plot of Power for Cycling of LPAC #2 with Lube Oil Heaters Energized. The events are annotated with letters and are explained in Table 3.12

Table 3.12: Events for LP Air Test Corresponding to Figure 3-20

Event	Description
A	Power due to cycling of LO Heaters
B	Effect of LPAC #2 Cycling visible.
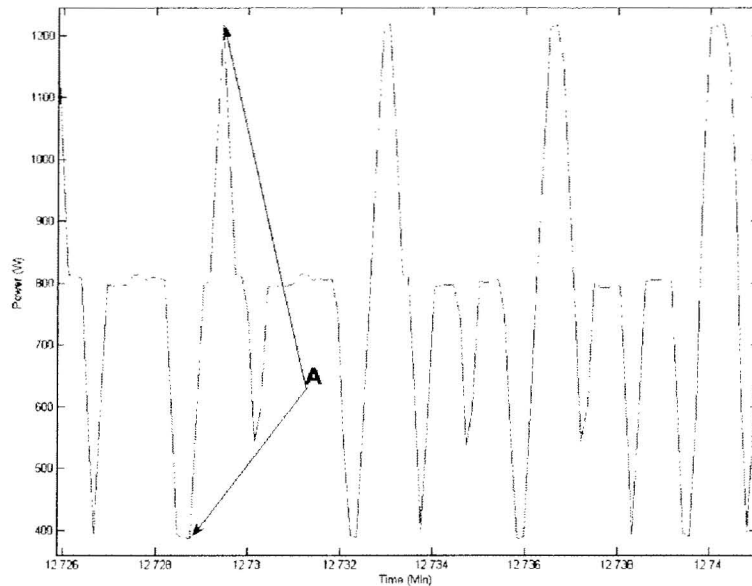
Figure 3-21: Close up of Power Trace for Cycling of Lube Oil Heaters. 'A' in Figure is the same as Figure 3-20

3.3.2   LP Air Tests

The power signals on LPAC #2 were noted to be abnormal by LCDR Bennett during his research [3]. Figure 3-22 shows the LPAC #2 power trace recorded by LCDR Greene in April 2005 and it is evident that the loading and unloading times of the compressor occur with a regular frequency during times of normal load. When a large load is placed on the system, such as when a GTM starts, LCDR Greene noted that the power trace indicated a variation in its normal pattern. By comparison Figure 3-23 shows the power trace of LPAC #2 in April 2007. The unloaded states of the compressor are no longer regular and show an increase in the unloaded times at irregular intervals. Figure 3-24 shows an enlarged portion of the power trace for LPAC #2 in April 2007 along with the LP Air Receiver pressure. This plot shows that the compressor unload time and the LP Air Receiver pressure are correlated, the plot also shows unusual behavior during certain unload intervals. The cause of the extended unload periods was unknown to LCDR Bennett but they were not found to relate to any plant operations.
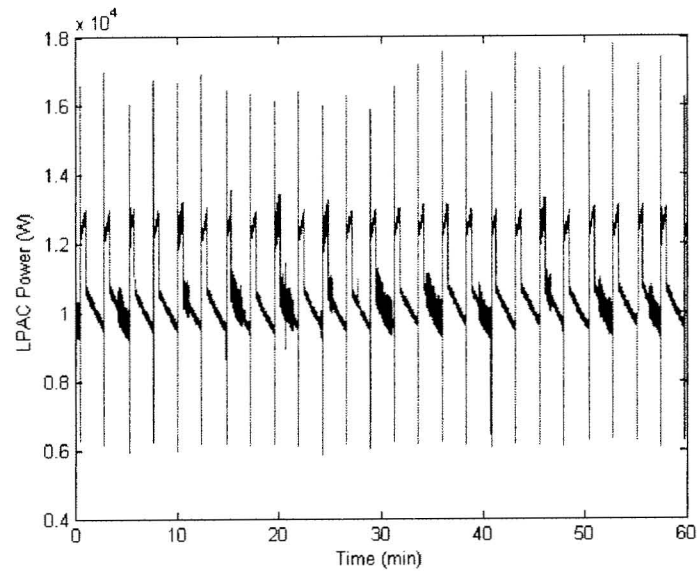
Figure 3-22: Plot of Power for LPACs #2 from April 2005. The load and unload times of the compressor are at a regular frequency during times of normal loading. [3]
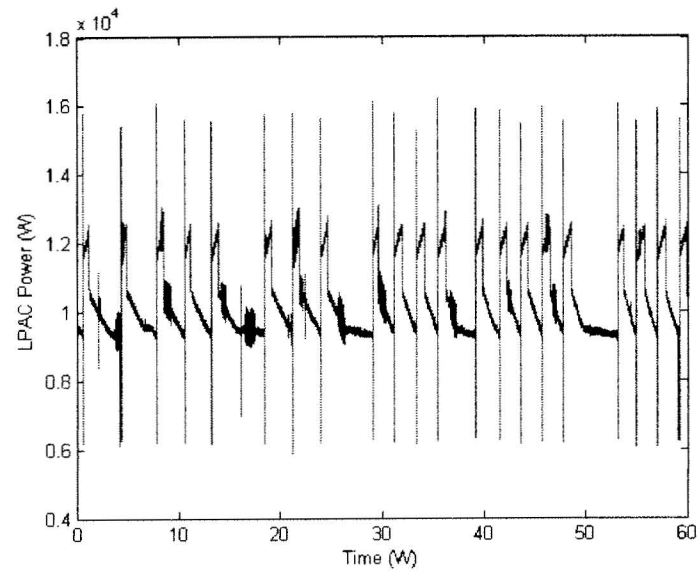
Figure 3-23: Plot of Power for LPACs #2 from April 2007. The unloaded times of the compressor are not regular and increase at irregular intervals. [3]

An LP Air test was conducted by operating #2 LPAC in automatic mode, first with the site air supply valve open then with it shut. The operation of the LPAC was observed during

51
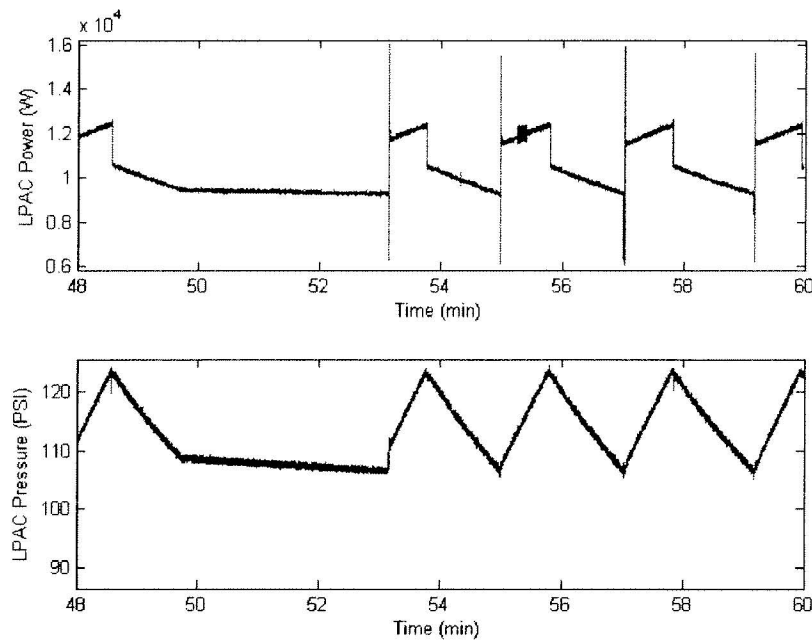
Figure 3-24: Close-up Plot of Power for LPACs #2 from April 2007. The unload times of the compressor are correlated to the LP Air System pressure. [3]

the test and the NILM data was analyzed upon completion. LPAC #2 NILM monitors a pressure transducer installed on the gauge line to the LP Air Receiver. The relative locations of components withing the LP Air system are shown on the LP Air System Diagram for LBES in Appendix B. The test was then repeated on LPAC #3 to confirm the results and to observe the cycling behavior of each LPAC. The LP Air Test procedure is included in Appendix C.

Figure 3-25 shows the results of the test on LPAC #2. The details of the events annotated in the figure are shown in Table 3.13. The upper trace shows the power signal to LPAC #2. A 25 SCFM leak was introduced into the system to provide an air load because normal plant loads were not operating during the test. Note that the LPAC exhibits the behavior with the excessive unload time when FA-V007 is open. The bottom trace shows the LP Air System pressure decreases to approximately 111 PSIG around 13 minutes into the test then increases to approximately 122 PSIG . It can be noted that the increase in pressure is not caused by the LPAC #2 as the power signal did not change during this time and the LPAC remained in its unloaded state. At about 15 mintues into the test the Site

Air Isolation Valve, FA-V007, was shut. The LPAC #2 power trace and the pressure trace indicate that the pressure cycles and LPAC load and unload states are now correllated. The LP Air System pressure increases during the LPAC load states and decreases during LPAC unload states. Approximately 28 minutes into the test FA-V007 was reopened and the LP Air System pressure returned to previous cycling and LPAC #2 returned to very short load states and excessively long unload states.

A second set of tests were conducted using LPAC #3. These tests also involved actuation of FA-V007. As shown in Figure 3-26 similar results were obtained. The details of the events annotated in the figure are shown in 3.14. It is easy to see that LPAC #3 has a diffferent operating power profile than #2 however, the same extended unload state is evident. Approximately 32 minutes into the test the operating pressure of LPAC #3 was changed to 110 to 120 PSIG to observe how the signals would change as monitored by the NILM system.

Conversations with an LBES technician indicated that the site air compressors had been replaced with larger, higher capacity compressors. Upon investigation, the cycling pressure range of the site air compressor was found to be 115 to 125 PSIG at the discharge of the compressor. The expected discharge pressure for the site air compressors is 90 PSIG. This would provide approximately 80 PSIG at the point where the site LP Air system taps into the LBES LP Air system, which is lower than the minimum cycling setpoint for both LPACs. With the site air compressors' operating band set to 115 to 125 PSIG, the pressure on the site side of check valve APL-V003 will exceed the pressure on the LBES side during normal cycling of the LBES LPACs. Therefore, the site air will supply the LBES LP Air system. The LBES LPACs will sense the pressure being maintained above their lower cycling setpoint and as a result will have long unload states. The LBES LPACs will not cycle on LP Air load as designed. There is no negative effect on the operation of LBES as a test facility or as a training environment for the ship's crews.
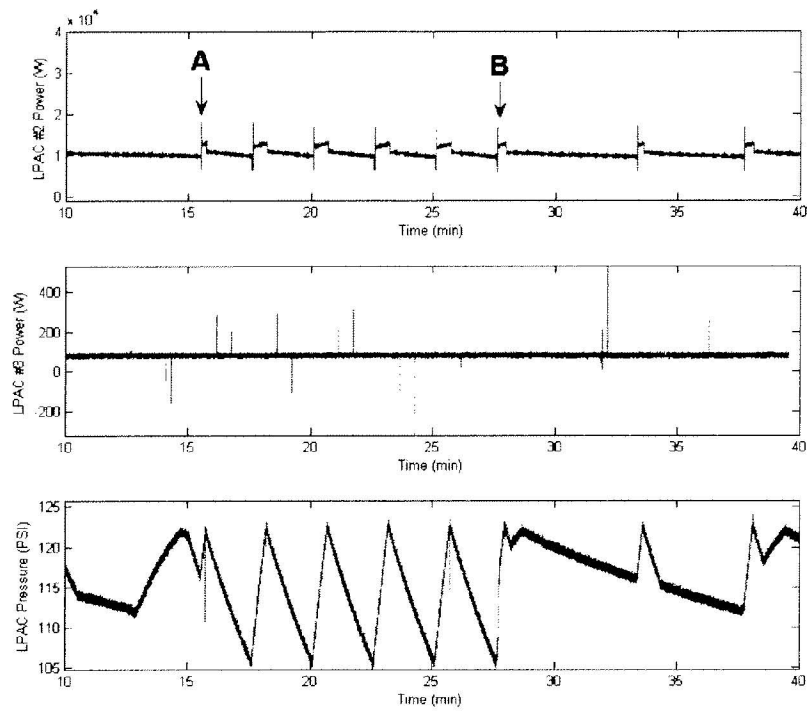
Figure 3-25: Plot of Power for LPACs #2 and #3 with LP Air Header Pressure. Upper trace: LPAC #2 power. Middle trace: LPAC #3 power, Bottom Trace: Receiver pressure. Label locations correspond to changes in system conditions. The events are annotated with letters and are explained in Table 3.13

Table 3.13: Events for LP Air Test Corresponding to Figure 3-25

Event	Description
A	Site LP Air Isolation valve FA-V007 shut
B	Site LP Air Isolation valve FA-V007 open
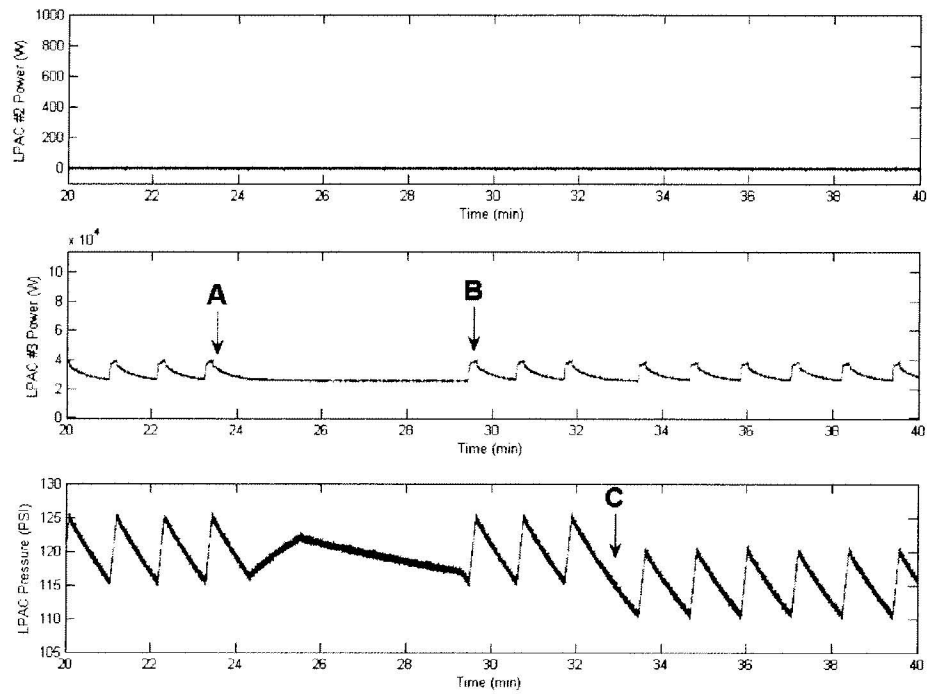
Figure 3-26: Plot of Power for LPACs #2 and #3 with LP Air Header Pressure. Upper trace: LPAC #2 power. Middle trace: LPAC #3 power, Bottom Trace: Receiver pressure. Label locations correspond to changes in system conditions. The events are annotated with letters and are explained in Table 3.14

Table 3.14: Events for LP Air Test Corresponding to Figure 3-26

Event	Description
A	Site LP Air Isolation valve FA-V007 open
B	Site LP Air Isolation valve FA-V007 shut
C	Output Pressure of #3 LPAC changed to 110-120 PSIG
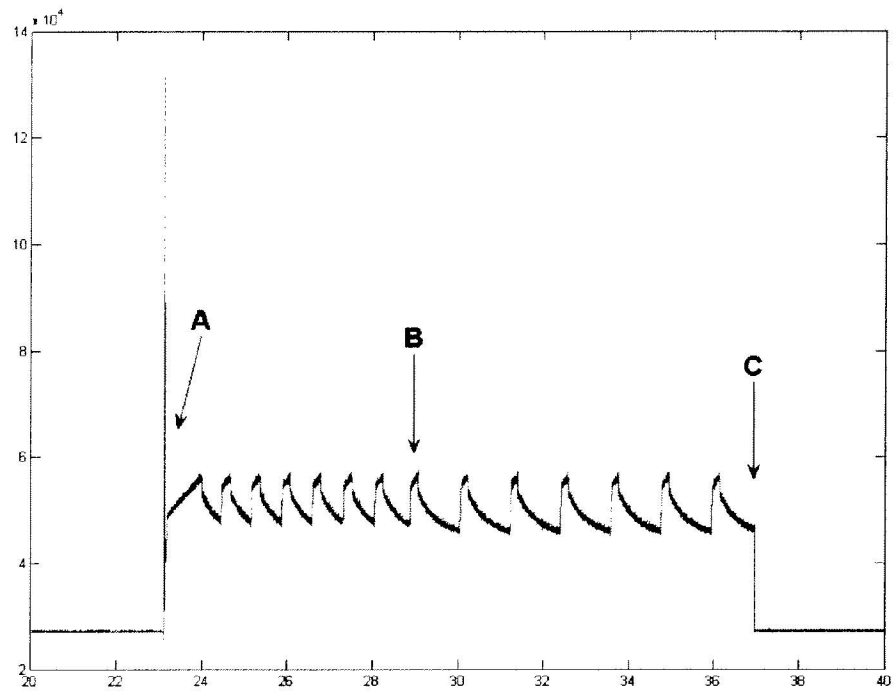
Figure 3-27: Plot of Power for a Decreased Air Load on LPAC #3. The events are annotated with letters and are explained in Table 3.15

Table 3.15: Events for LP Air Test Corresponding to Figure 3-27

Event	Description
A	LPAC #3 on in automatic at 115 to 125 PSIG
B	Decreased air load on LP Air System
C	LPAC #3 Secured

Chapter 4

Multi-Function Monitor (MFM)

4.1 Introduction

Modern surface warships employ some form of Zonal Electrical Distribution System (ZEDS) to increase the reliability, quality of service [15] and survivability or the ship in the event the system is damaged [16, 33]. The U.S. Navy has plans to develop and deploy an Integrated Power System (IPS) for future surface warships. IPS systems combine the electrical power generation and propulsion power systems [13]. The MFM is a central component of these systems to provide switching and fault isolation [6].

4.2 MFM System Description

In naval shipboard electrical distribution systems prior to utilizing a ZEDS, fault isolation was performed by selecting the overcurrent settings on circuit breakers such that those closest to the fault will trip first. This process is called selective tripping [27]. If the system works as designed, the fault will be isolated while minimizing the impact on the rest of the system. Various studies, analyses and tests have shown that selective tripping may not adequately protect the electrical distribution system during battle damage if the damage is not a single point failure [33]. With the implementation of ZEDS, the situation was further complicated because the direction of current flow through the components in the system depends on the system configuration. This necessitated a protective system which could sense the direction of current flow and provide intelligent fault detection and isolation actions [33]. The ZEDS system for a DDG-51 Flight IIA Arleigh Burke class destroyer is

shown in Figure 4-1.
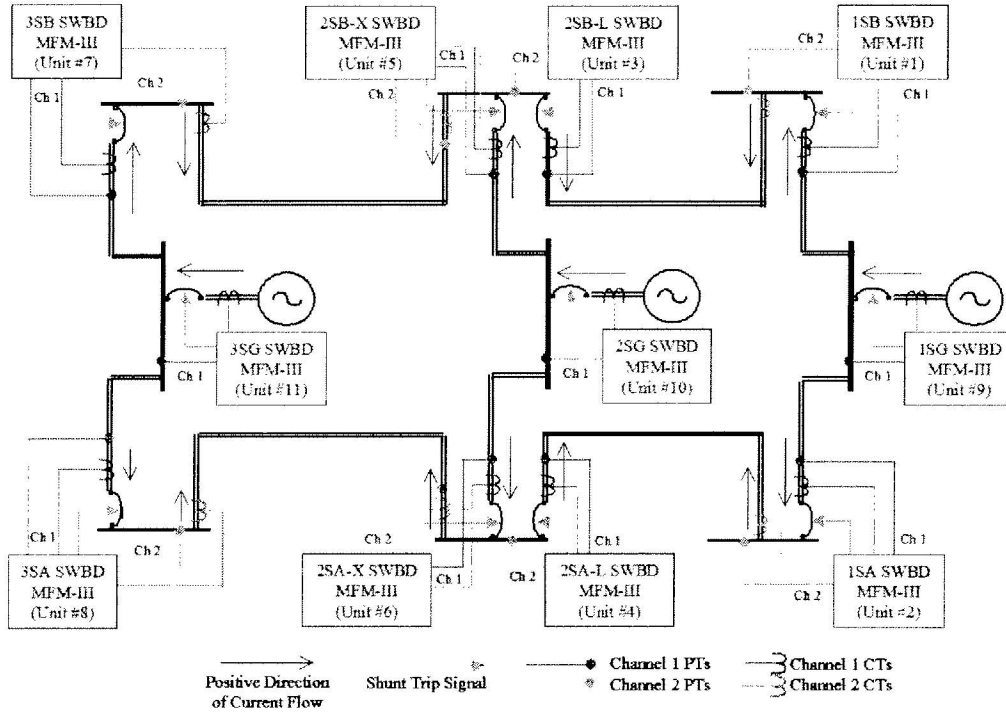
Figure 4-1: DDG-51 Flight IIA ZEDS [6] This diagram shows the addressing, locations and signal inputs of the MFMs located in the system. Also shown is the defined positive direction of current flow for each MFM.

The MFM-I was designed to accomplish this task. The MFM-III is the improved version of the MFM-I and is discussed in this thesis. Each MFM-III has two sets of sensors, one located on each side of its associated circuit breaker. The exception to this are the generator output breaker which utilize a single set of sensors. There are two line voltage measurements: v_{ab} and v_{bc} from two potential transformers (440:110), and three phase currents: i_a, i_b and i_c from current transformers (CTs) (6000:5). The voltage and current inputs are used to calculate the power flowing through the circuit breaker. The CTs provide an indication of the direction of current flow. The trip status of the associated circuit breaker is also an input to the MFM logic circuit [6]. A functional diagram of the MFM is shown in Figure 4-2.

There are four types of MFMs based on where they are to be placed in the distribution system. In order for the software to be able to determine the correct system status and fault determination, each MFM is assigned a unique identifier which indicates the specific location in the system where it is located. To provide a means for the MFMs to pass data

to the other MFMs in the system, each MFM is equipped with three ethernet ports. Port #1 is used to transmit and receive data on the ring connection with all other MFMs. Data is transmitted through port #1 every 5.0 ms. Ports #2 & #3 are used to transmit and receive data point-to-point with the MFMs on either side. Data is transmitted through these ports every 1.0 ms. This ability of the MFMs to exchange data throughout the entire system allows for a coordinated response to a detected fault. The system is designed to initiate a trip response within 10 ms of detection [6].
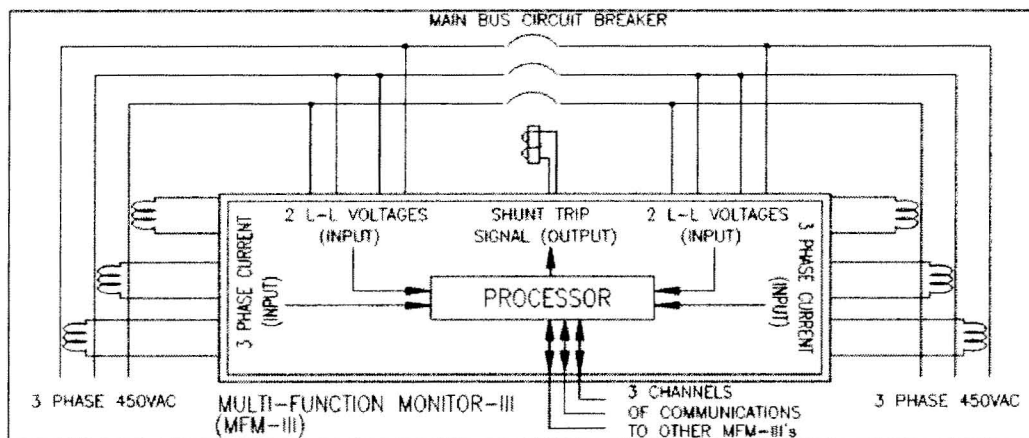
Figure 4-2: Functional Diagram of MFM [6] The diagram shows the sensor layout for the MFM. The three phase current and two voltage measurements per channel, the three ethernet communications channels and the shunt trip output to the associated circuit breaker. The shunt trip status input of the circuit breaker is not shown in the diagram.

4.3 MFM System Theory of Operation

There are two main software routines that the MFM-III uses. The first is called the High Speed Relay (HSR) and was developed by Barrons Associates Inc (BAI). The HSR routine is used to determine if certain faults exist and to calculate power and current levels. These values are inputs to the second routine, the Integrated Protective Coordination System (IPCS). IPCS was developed by the Naval Surface Warfare Center, Carderock Division Code 982 with assistance from Micheal Sieleman at Bath Iron Works (BIW) [6]. The inputs to the HSR routine are similar to those used by NILM. Therefore, the HSR routine will be looked at in detail.

4.3.1 The HSR Algorithm

The HSR routine contains three main routines: the Park's transformation, a fault detection algorithm, and fault direction determination. There are five outputs of the HSR Algorithm that are used by the IPCS Algorithm: Angle and Magnitude (from Park's Transformation), the change in the Angle, Instantaneous Power, and Fault Direction. The details of these routines and outputs are discussed below in this section.

Park's Transformation

A Park's transformation is used to change the reference frame from that of the physical stator to that of the theoretical rotating 3-phase electrical signal . The general Park's Transformation matrix that is used in the HSR algorithm is

$$
\begin{bmatrix} F_d \\ F_q \\ F_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos\left(\theta\right) & cos\left(\theta - \frac{2\pi}{3}\right) & cos\left(\theta + \frac{2\pi}{3}\right) \\ sin\left(\theta\right) & sin\left(\theta - \frac{2\pi}{3}\right) & sin\left(\theta - \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix}. \tag{4.1}
$$

where f can represent voltage or current [20]. The Park's Transformation is normally performed using all three values for f but the MFM only measures two line voltages, therefore the equations must be modified to reflect the measurements taken. A useful relation of a 3-phase electrical circuit is that the sum of the line voltages must be equal to zero,

$$
V_{ab} + V_{bc} + V_{ca} = 0. \tag{4.2}
$$

Looking at the first row of the Park's Transformation to compute F_d using the line voltages for f yields

$$
V_d = V_{ab}cos\theta + V_{bc}cos\left(\theta - \frac{2\pi}{3}\right) + V_{ca}cos\left(\theta + \frac{2\pi}{3}\right). \tag{4.3}
$$

Using the relationship of Eq. (4.2) and substituting for V_{ca} gives

$$
V_d = V_{ab}cos\theta + V_{bc}cos\left(\theta - \frac{2\pi}{3}\right) - (V_{ab} + V_{bc})cos\left(\theta + \frac{2\pi}{3}\right). \tag{4.4}
$$

Multiplying through and gathering like terms

$$
V_d = V_{ab}\left[-2sin\left(\frac{2\theta + \frac{2\pi}{3}}{2}\right)sin\left(\frac{\theta - \theta - \frac{2\pi}{3}}{2}\right)\right]. \tag{4.5}
$$

60

Using the trigonometric identity

$$cos\,u - cos\,v = -2sin\left(\frac{u+v}{2}\right)sin\left(\frac{u-v}{2}\right) \tag{4.6}$$

and substituting into Eq. (4.5)

$$
\begin{aligned}
V_d &= V_{ab}\left[cos\theta - cos\left(\theta + \frac{2\pi}{3}\right)\right] + V_{bc}\left[cos\left(\theta - \frac{2\pi}{3}\right) - cos\left(\theta + \frac{2\pi}{3}\right)\right] \\
&\quad + V_{bc}\left[-2sin\left(\frac{2\theta}{2}\right)sin\left(\frac{\theta - \theta - \frac{4\pi}{3}}{2}\right)\right]..
\end{aligned} \tag{4.7}
$$

Simplifying gives

$$V_d = 2V_{ab}\left[sin\left(\theta + \frac{\pi}{3}\right)sin\left(\frac{\pi}{3}\right)\right] + 2V_{bc}\left[sin\theta\,sin\left(\frac{2\pi}{3}\right)\right]. \tag{4.8}$$

Similarly, for the second row of the Park's Transformation to compute F_q using the line voltages for f yields

$$V_q = V_{ab}sin\theta + V_{bc}sin\left(\theta - \frac{2\pi}{3}\right) + V_{ca}sin\left(\theta + \frac{2\pi}{3}\right). \tag{4.9}$$

Using the relationship of Eq. 4.2 and substititing for V_{ca} gives

$$V_d = V_{ab}sin\theta + V_{bc}sin\left(\theta - \frac{2\pi}{3}\right) - (V_{ab} + V_{bc})\,sin\left(\theta + \frac{2\pi}{3}\right). \tag{4.10}$$

Multiplying through and gathering like terms

$$V_d = V_{ab}\left[sin\theta - sin\left(\theta + \frac{2\pi}{3}\right)\right] + V_{bc}\left[sin\left(\theta - \frac{2\pi}{3}\right) - sin\left(\theta + \frac{2\pi}{3}\right)\right]. \tag{4.11}$$

Using the trigonometric identity

$$sin\,u - sin\,v = 2cos\left(\frac{u+v}{2}\right)sin\left(\frac{u-v}{2}\right) \tag{4.12}$$

and substituting into Eq. (4.11)

$$V_d = V_{ab}\left[2cos\left(\frac{2\theta + \frac{2\pi}{3}}{2}\right)sin\left(\frac{\theta - \theta - \frac{2\pi}{3}}{2}\right)\right]$$

$$+V_{bc}\left[2cos\left(\frac{2\theta}{2}\right)sin\left(\frac{\theta-\theta-\frac{4\pi}{3}}{2}\right)\right]. \tag{4.13}$$

Simplifying gives

$$V_d = -2V_{ab}\left[cos\left(\theta+\frac{\pi}{3}\right)sin\left(\frac{\pi}{3}\right)\right] - 2V_{bc}\left[cos\theta sin\left(\frac{2\pi}{3}\right)\right]. \tag{4.14}$$

Noting that $2sin\left(\frac{\pi}{3}\right) = 2sin\left(\frac{2\pi}{3}\right)$ Eq. (4.8) and Eq. (4.14) can be further simplifed as follows

$$V_d = 2sin\left(\frac{\pi}{3}\right)\left[V_{ab}sin\left(\theta+\frac{\pi}{3}\right)+V_{bc}sin\theta\right] \tag{4.15}$$

and

$$V_q = -2sin\left(\frac{\pi}{3}\right)\left[V_{ab}cos\left(\theta+\frac{\pi}{3}\right)+V_{bc}cos\theta\right]. \tag{4.16}$$

The F_0 value is not calculated in the HSR routine. This value checks the three voltage inputs to verify that they sum to zero. Since only two line voltages are measured, the system is assumed to be a balanced 3-phase system and it is assumed that the sum of the voltages equals zero [18]. Once the F_q and F_d values are calculated, they are used to compute the magnitude and angle of the phasor

$$Magnitude = \frac{2}{3}\sqrt{F_q^2 + F_d^2} \tag{4.17}$$

$$Angle = \arctan\left(\frac{F_q}{F_d}\right). \tag{4.18}$$

Fault Detection

The Fault Detection routine uses two different tests to determine if a fault condition exists, a large change in the current angle computed by the Park's Transformation from the average angle from past eight samples and comparing the magnitude from the Park's Transformation to high and low thresholds.

Angle Fault The Δ_{Angle} is calculated as the change in the angle computed in the Park's Transformation from the average of the past eight valid samples. If a large change in the angle is detected, it is an indication of a severe increase in power and/or a reversal of the power flow. Both of these conditions are indicators that a fault condition may exist. If the calculated angle and the average angle differ in sign, the angle buffer is *unwrapped* by

adding π to the angles that are in the buffer so that all of the angles are in the quadrants I and IV. The absolute value of the Δ_{Angle} is compared to a threshold value. If it is exceeded the fault flag is set to "Angle Fault."

Magnitude Fault   The newly calculated magnitude is compared to low and high thresholds to verify if it is within the acceptable band of what is expected. If it is outside this band, then it is an indication that a severe increase in power flow, which is an indication that a fault condition exists. The fault flag is set to "Magnitude Fault."

Fault Direction

If the Fault Detection algorithm indicates that a fault exists, the power is calculated through both input channels to determine if the fault is upstream or downstream of the MFM. The CTs are oriented specifically during installation such that positive power values indicate current is flowing from upstream to downstream of the MFM. Therefore the power calculation will provide an indication of the direction of the electrical energy flowing through the circuit breaker.

Calculate Average Power   By definition the instantaneous power is

$$ P = v_a i_a + v_b i_b + v_c i_c. \tag{4.19} $$

The MFM does not monitor v_a, v_b, or v_c but rather v_{ab} and v_{bc}. Therefore the power must be computed in terms of the actual measured voltages. This can be accomplished by noting that the relationship between phase voltages v_a and v_b and line voltage v_{ab} is

$$ v_{ab} = v_a - v_b. \tag{4.20} $$

Similarly for v_{bc} and v_{ca}

$$ v_{bc} = v_b - v_c \tag{4.21} $$

and

$$ v_{ca} = v_c - v_a \tag{4.22} $$

Substituting Eq. (4.20) and Eq. (4.21) into Eq. (4.19) yields

$$P = (v_{ab} + v_b) i_a + (v_{bc} + v_c) i_b + v_c i_c. \tag{4.23}$$

Noting that the current for a 3-phase electrical system must satisfy the relation

$$i_a + i_b + i_c = 0 \tag{4.24}$$

and solving for i_b, this can be substituted into Eq. (4.23) resulting in

$$P = (v_{ab} + v_{bc} + v_c) i_a + (v_{bc} + v_c) (-i_a - i_c) + v_c i_c. \tag{4.25}$$

Gathering like terms

$$P = v_{ab} i_a + v_{bc} i_a + v_c i_a - v_{bc} i_a - v_{bc} i_c - v_c i_a - v_c i_c + v_c i_c. \tag{4.26}$$

Simplifying this equation gives

$$P = v_{ab} i_a - v_{bc} i_c. \tag{4.27}$$

This gives an accurate calculation of power in the circuit assuming that the loads are equally balanced. If there is a fault on phase B, the power would not be accurate since i_b is not included in the equation. To eliminate uncertainties and provide an indication of a single phase fault a method must be used such that all measurements receive equal weight. Noting that Eq. (4.27) was arrived at by substituting for i_b in Eq. (4.23), two more equations can be found by substituting for i_a and i_c. Solving for i_a in Eq (4.24) and substituting into Eq (4.23) results in

$$P = (v_{ab} + v_{bc} + v_c) (-i_b - i_c) + (v_{bc} + v_c) i_b + v_c i_c. \tag{4.28}$$

Simplifying the equation results in

$$P = -v_{ab} i_b - v_{ab} i_c - v_{bc} i_c. \tag{4.29}$$

64

Gathering like terms results in

$$P = -v_{ab} \left(i_b + i_c \right) - v_{bc} i_c. \tag{4.30}$$

Similarly, solving Eq (4.24) for i_c and substituting in to Eq (4.23) results in

$$P = v_{ab} i_a + v_{bc} i_a + v_c i_a + v_{bc} i_b + v_c i_b - v_c i_a - v_c i_b. \tag{4.31}$$

Simplifying the equation results in

$$P = v_{ab} i_a + v_{bc} i_a + v_{bc} i_b. \tag{4.32}$$

Gathering like terms results in

$$P = v_{ab} i_a + v_{bc} \left(i_a + i_b \right). \tag{4.33}$$

Each of these three equations for power compute the power for the entire circuit. To arrive at an accurate power computation where all three phase currents are accounted for equally, the average of Equations (4.27), (4.30), and (4.33) is calculated. Therefore, the instantaneous power in the circuit including all three phase currents is

$$Power = \left[\left(v_{ab} i_a - v_{bc} i_c \right) + \left(v_{bc} \left(i_a + i_b \right) + v_{ab} i_a \right) + \left(-v_{bc} i_c - v_{ab} \left(i_b + i_c \right) \right) \right] / 3 \tag{4.34}$$

Determine Fault Direction If no fault is detected by the fault detection routine, the power value is added to a buffer of the last 16 power values with no faults detected. This buffer will be used to calculate the average power which will be used for fault direction determination when a fault is detected.

If a fault is detected, the calculated power is subtracted from the average of the buffer to determine a Δ_{Power} as shown in Equation 4.35. The equation used calculate Δ_{Power} is

$$\Delta_{Power} = \frac{\Sigma Power}{16} - Power. \tag{4.35}$$

If the Δ_{Power} is positive, indicating an decrease or reversal in electrical energy flowing through the circuit breaker, the Fault Direction flag is set to Upline, otherwise it is set to

Downline.

4.3.2 Example of MFM HSR Algorithm

To illustrate the operation of the HSR algorithm, the following examples are presented. The values for current are assumed values and are given only to provide an indication of the changes in magnitudes of the current at various points in the circuit. Figure 4-3 shows the circuit at some instant of time with two ideal power sources and three loads at a steady-state condition with no faults in the circuit. The three loads are each supplied from a separate switchboard. The circuit breakers between the switchboards are each monitored by separate MFMs, each with two channels monitoring either side of the circiut breaker. The table shows the output variables and how they change between no fault and fault conditions. Arrows are used to give an indication of the relative change to the steady-state values. With no faults in the circuit, the fault direction determination is not performed, therefore "N/A" is entered into the table.
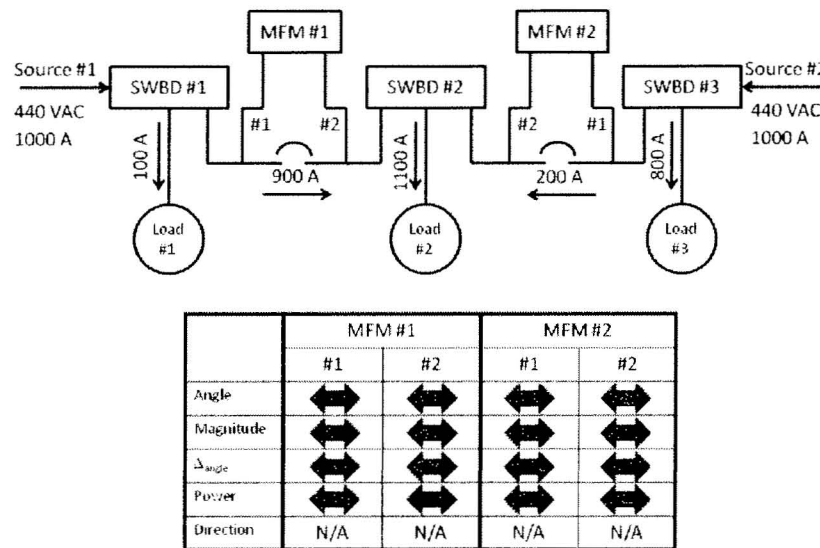
Figure 4-3: MFM HSR Algoritm Example: Steady State Conditon. Two ideal power sources supply the loads off of the switchboards in a steady-state condition with no faults in the circuit.

Figure 4-4 illustrates what outputs would be from the MFM HSR algorithms if a fault exists on Switchboard #1. In this simple example, the current is assumed to flow almost

entirely to the fault, therefore the load currents decrease to near zero. As can be seen, the current direction through MFM #1 is reversed from steady-state and has increased in magnitude. The Park's Transformation for MFM #1 will result in an increase in magnitude and a large change in the angle calculation (it is shown negative assuming that the steady-state vector was in quadrant I and the reversal of the current cause the shift of the vector towards quadrant IV). The large increase in the magnitude and the large change in the angle will cause both fault flags to be set, therefore the direction determination will be performed. Since the current flow is reversed through MFM #1 with no change in the voltage phase, the instaneous power calculated will also be negative. Calculating Δ_{Power} using Eq. 4.35 results in a positive value which indicates the fault is "upline" or upstream of the MFM #1 in relation to the defined *normal* power flow.

The fault causes the power through MFM #2 to increase. If the fault is large enough, the maginitude and angle flags will be set due to the large change in the associated values. Therefore the direction determination will also be performed in MFM #2. Since the power only increased and didn't reverse, Δ_{Power} will be negative, indicating to the MFM that the fault is "downline" or downstream from MFM #2 in relation to the defined *normal* power flow.

Figure 4-5 illustrates a fault on Switchboard #2. The outputs for this scenario for each MFM will be similar to those for MFM #2 for the fault on Switchboard #1. For MFM #2, it will see no difference between the two fault locations, both are downstream. For MFM #1, the power will not reverse, therefore it will also indicated that the fault is downstream. The magnitude, angle, Δ_{Angle}, and power will increase on both MFMs.

The last scenario presented is that of a fault on MFM #2 itself, between the sensor locations. Figure 4-6 shows the system diagram and the output table for this scenario. MFM #1 will see the same indications as the fault on Switchboard #2 and will indicate the fault is downstream. MFM #2 will not show the same indications on both of its sensor taps. For the #1 tap, the power will increase in the same direction as steady-state, therefore tap #1 will indicate that the fault is downstream. The power flow through tap #2 will reverse, and similar to MFM #1 in the scenario with the fault on Switchboard #1, tap #2 will indicate that the fault is upstream.
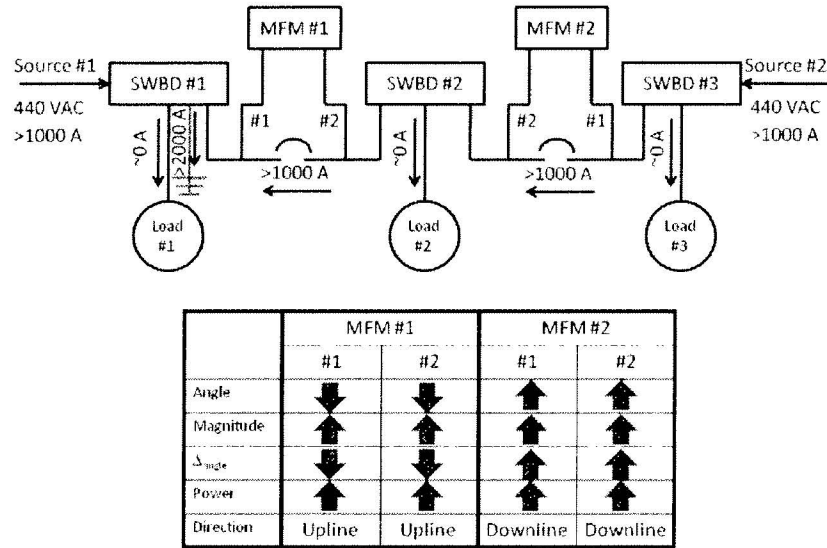
Figure 4-4: MFM HSR Algoritm Example: Fault on Switchboard #1. The fault causes the power flow through MFM #1 to reverse. MFM #1 indicates that the fault is UPLINE. The fault causes an increase in the power flow through MFM #2 in the same direction as steady-state. MFM #2 indicates that the fault is DOWNLINE.

4.4 Integration of NILM with the MFM

The MFM system uses similar input signals to NILM and both systems have similar basic functionality, to monitor the electrical power being used by the electrical loads in the system. The main focus of the NILM system is to determine detailed operating characteristics and conditions of the equipment. The main focus of the MFM system is for protection of the electrical system during fault conditions and to isolate faults such that as much of the distribution system is maintained in an operable condition as possible.

The MFM system must operate very rapidly to be effective in fault isolation. One area where the NILM system could provide a useful benefit is to maintain state information on major loads in the electrical distribution system. This state information will provide two benefits to the operation of the MFM. First, if all of the largest loads in the system are running and a large increase in power is detect by the MFM, the MFMs could take action more quickly because time would not have to be taken to determine if the increase in power is a piece of equipment or a fault. Secondly, the actual current setpoints could be dynamically set based on the operating conditions of the plant equipment.

68
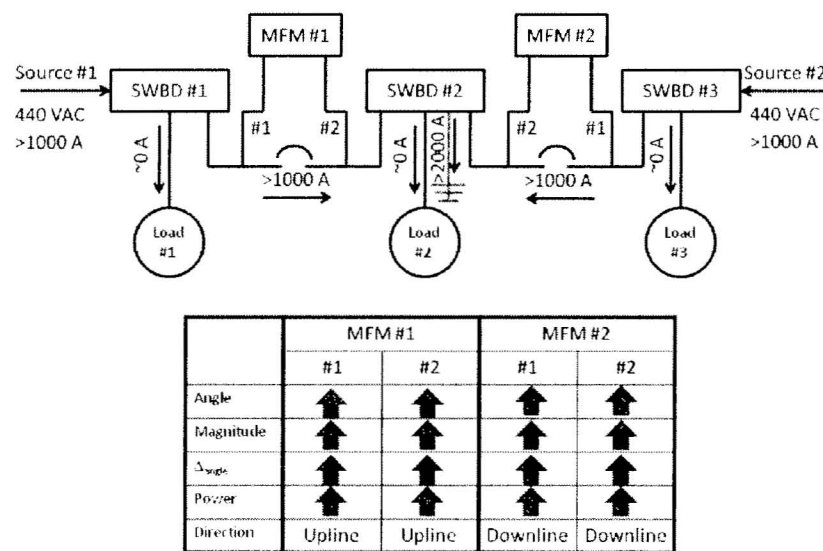
Figure 4-5: MFM HSR Algoritm Example: Fault on Switchboard #2. The fault causes an increase in the power flow through MFMs #1 & #2 in the same direction as steady-state. MFMs #1 & #2 indicate that the fault is DOWNLINE.
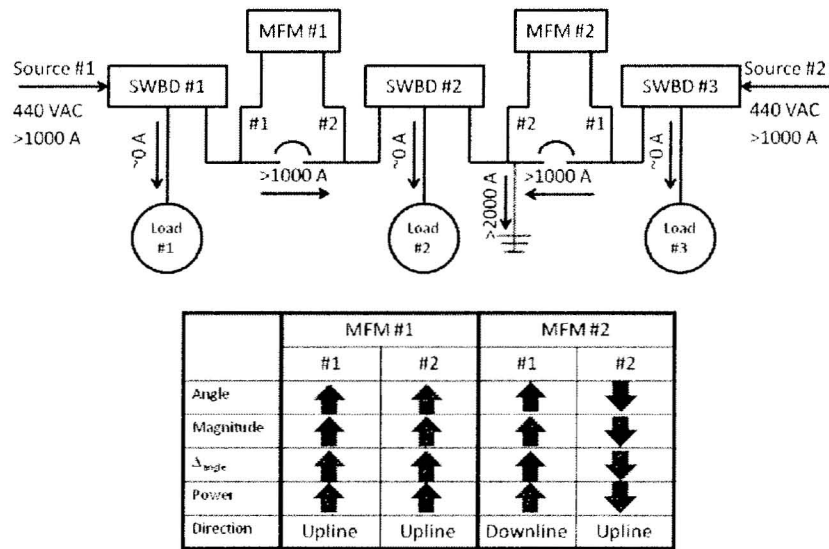
Figure 4-6: MFM HSR Algoritm Example: Fault on MFM #2. The fault causes an increase in the power flow through MFM #1 in the same direction as steady-state. MFM #1 indicates that the fault is DOWNLINE. MFM #2 tap #1 senses an increase in power flow in the same direction as steady-state, therfore tap #1 indicates that the fault is DOWNLINE. MFM #2 tap #2 senses increase in power flow opposite in direction to steady-state, therfore tap #2 indicates that the fault is UPLINE

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The NILM system has proven to be able to provide useful data in the monitoring of shipboard systems. Through the use of a GUI the shipboard crews will gain the ability to operate the system during operation, providing them with diagnostic indicators and system status while the shipboard systems are in operation.

The 1SA and 3SA NILM systems installed at LBES have shown in preliminary testing to be able to distinguish the individual componenet signals from the aggregate signal of all loads. Further testing is required when the LBES facility is in full operation to assess the ability of the 1SA & 3SA NILMs to distinguish the individual equipment operating characteristics in the aggregate signals.

The cause of the abnormal operation of the LPACs at LBES was determined to be the installation of new site air compressors which are set to operate in the normal pressure ranges of the LBES LPACs. The site air compressors will supply the LP Air loads of LBES whenever their pressure exceeds that of the LBES LPACs.

5.2 Future Work

5.2.1 LBES

There is a need to further explore utilizing the aggregate signals at the LBES facility to characterize the operating states of the equipment. This analysis can be compared to the states determined by the NILMs installed on the individual equipment. Automated routines

could be developed to provide realtime indication of plant conditions. This would facilitate the NILM being incorporated into the Next Generation Integrated Power System (NGIPS) on future Navy ships.

5.2.2   NILM GUI

Further work is needed to implement a simple GUI for the NILM systems installed aboard ships. The crews will need to have something they can use without requiring extensive training. The information presented should be informative and the operation of the GUI should be intuitive.

Bibliography

[1] CDR John V. Amy Jr. USN. Considerations in the Design of Naval Electric Power Systems. In *Power Engineering Society Summer Meeting, 2002 IEEE*, volume 1, pages 331–335, Chicago, IL, USA, July 2002. IEEE.

[2] CDR John V. Amy Jr. USN (Ret.) Ph.D. P.E. Modern, High-Converter-Populations Argue for Changing How To Design Naval Electric Power Systems. Technical report, BMT Syntek Technologies, USA, 2005.

[3] Patrick Lawrence Bennett. Using the Non-Intrusive Load Monitor for Shipboard Supervisory Control. Master's thesis, Massachusetts Institute of Technology, 2007.

[4] Barry W. Boehm. A Spiral Model of Software Development and Enhancement. *Computer*, 21(5):61–72, 1988.

[5] R. J. Bost, J. G. Mellis, and P. A. Dent. Is the Navy Serious About Reducing Manning on its Ships? *Office of Naval Research*, 1999.

[6] NSWCCD-SSES Code98/9345. A Guide To MFM III Operation (DRAFT). Technical report, NAVSEA Naval Surface Warfare Center Carderock Division Naval Ships Systems Engineering Station, Philadelphia, Pennsylvanina, 2003.

[7] R. W. Cox. *Minimally Intrusive Strategies for Fault Detection and Energy Monitoring.* PhD thesis, MIT, Aug 2006.

[8] R. W. Cox, G. Mitchell, P. Bennett, M. Piber, J. Paris, W. Wichakool, and S. B. Leeb. Improving Shipboard Maintenance Practices Using Non-Intrusive Load Monitoring. In *Proceedings ASNE Intelligent Ships VII, Philadelphia, PA*.

[9] R. W. Cox, J. P. Mosman, D. McKay, S. B. Leeb, and T. J. McCoy. Diagnostic Indicators for Shipboard Cycliing Systems Using Non-Intrusive Load Monitoring. ASNE Day 2006, 2006.

[10] T. DeNucci, R. W. Cox, S. B. Leeb, J. Paris, T. J. McCoy, C. R. Laughman, and W. C. Greene. Diagnostic Indicators for Shipboard Systems Using Non-Intrusive Load Monitoring. In *Proceedings IEEE Electric Ship Technologies Symposium, Philadelphia, PA*.

[11] CAPT Norbert Doerry. Zonal Ship Design. *American Society of Naval Engineers Naval Engineer's Journal*, 118(1):39–53, Winter 2006.

[12] CAPT Norbert Doerry. Designing Electrical Power Systems for Survivability and Quality of Service. *American Society of Naval Engineers Naval Engineer's Journal*, 119(2):25–34, 2007.

[13] CAPT Norbert Doerry. Next Generation Integrated Power System NGIPS Technology Development Roadmap. Technical report, Naval Sea Systems Command, November 2007.

[14] CAPT Norbert H. Doerry. Systems Engineering and Zonal Ship Design. Technical report, Naval Sea Systems Command, May 2006. Presented at the ASNE Engineering the Total Ship Symposium, ETS 2006, May 1-2, 2006.

[15] CAPT Norbert H. Doerry and David H. Clayton. Shipboard Electrical Power Quality of Service. Technical report, IEEE, July 2005. Presented at the IEEE Electric Ship Technologies Symposium, Philadelphia, PA, July 25-27, 2005.

[16] CAPT Norbert H. Doerry and Howard Fireman. Designing All Electric Ships. In *Proceedings fo the Ninth International Marine Design Conference*, May 2006.

[17] ENVIROVAC INC., 1260 Turret Drive, Rockford, Illinois 61111. *Instruction Manual for the Envirovac Vacuum Collection System, United States Coast Guard 270' B Class WMEC, Technical Publications Number 2839*, April 1983.

[18] William C. Greene. Evaluation of Non-Intrusive Load Monitoring for Condition Based Maintenance Applications of U.S. Navy Propulsion Plants. Master's thesis, MIT, 2005.

[19] Henry Hegner and Bipin Desai. Integrated Fight Through Power. Technical report, IEEE, 2002.

[20] Paul C. Krause, Oleg Wasynczuk, and Scott D. Sudhoff. *Analysis of Electric Machinery and Drive Systems*. Piscataway, NJ : IEEE Press ; New York : Wiley Interscience, c2002., 2nd edition, 2002.

[21] Mark W. Maier and Eberhardt Rechtin. *The Art of Systems Architecting, Second Edition*. CRC Press, 2002.

[22] Timothy J. McCoy. Manning and Automation. M IT Power Point Presentation, March 2007.

[23] Thomas Duncan McKay. Diagnostic Indicators for Shipboard Mechanical Systems Using Non-Intrusive Load Monitoring. Master's thesis, Massachusetts Institute of Technology, 2006.

[24] J. P. Mosman. Evaluation of Non-Intrusive Load Monitoring for Shipboard Cycling System Diagnostics. Master's thesis, Massachusetts Institute of Technology, June 2006.

[25] Naval Sea Systems Command. *Technical Manual for AG9140RF Ship Service Gas Turbine Generator (SSGTG), Description, Operation, and Scheduled Maintenance, S9311-XX-MMO-010*, vol 1 edition.

[26] Naval Sea Systems Command. *Technical Manual for Air Compressor Low Pressure, Oil Free, S6226-DE-MMA-010/07524*, November 1992.

[27] Naval Sea Systems Command. *Naval Ships' Technical Manual Chapter 320, Electrical Power Distribution Systems*, revision 2 edition, April 1998.

[28] Naval Sea Systems Command. *Technical Manual for Machinery Control System, Shaft Control Unit, P/N 63E915603G1, S9202-AN-MMF-020*, volume 2 edition, Oct 1998.

[29] Naval Sea Systems Command. *Technical Manual Organizational Level Maintenance LM2500 Propulsion Gas Turbine Module Description and Operation, S9234-AD-MMO-010/LM2500*, revision 3 edition, May 2000.

[30] Naval Sea Systems Command. *Compressor, Air, Low Pressure, Oil-Free, Model Star 200, S6220-EE-MMA-010*, change f edition, October 2002.

[31] Naval Sea Systems Command. *Unverval Engine Controller (UEC) Operations and Maintenance, S6263-B7-MMA-010*, vol 1 & vol 2, change a edition, Aug 2003.

[32] Mark A. Piber. Improving Shipboard Maintenance Practices Using Non-Intrusive Load Monitoring. Master's thesis, Massachusetts Institute of Technology, 2007.

[33] Lyle Plesnick, Tracy Hannon, and Daniel Devine. An Intelligent Fault Detection Device For Shipboard Power Systems. Technical report, Naval Surface Warfare Center, Carderock Division / U.S. Navy.

[34] Ethan Proper. Automated Classification of Power Signals. Master's thesis, Massachusetts Institute of Technology, June 2008.

[35] J. S. Ramsey, S. B. Leeb, T. DeNucci, J. Paris, M. Obar, R. W. Cox, C. R. Lauhman, and T. J. McCoy. Shipboard Applications of Non-Intrusive Load Monitoring. In *Proceedings ASNE Reconfiguration and Survivability Symposium, Atlandic Beach, FL*, 2005.

[36] Jack S. Ramsey. Shipboard Applications of Non-Intrusive Load Monitoring. Master's thesis, Massachusetts Institute of Technology, 2004.

[37] S. R. Shaw. *System Indentification Techniques and Modeling for Non-Intrusive Load Diagnostics*. PhD thesis, MIT, February 2000.

[38] *Systems Engineering, Principles and Practice*. Wiley Interscience, 2003.

Appendix A

NILM Transducer Data Sheets

Voltage Transducer LV 25-P

For the electronic measurement of voltages : DC, AC, pulsed...,
with a galvanic isolation between the primary circuit (high voltage)
and the secondary circuit (electronic circuit).

I_{PN} = **10 mA**
V_{PN} = **10 .. 500 V**

CE
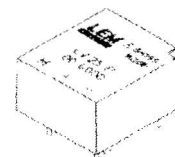
Electrical data

I_{PN}	Primary nominal r.m.s. current		10		mA
I_P	Primary current, measuring range		0 .. ± 14		mA
R_M	Measuring resistance		$R_{M min}$	$R_{M max}$	
	with ± 12 V	@ ± 10 mA $_{max}$	30	190	Ω
		@ ± 14 mA $_{max}$	30	100	Ω
	with ± 15 V	@ ± 10 mA $_{max}$	100	350	Ω
		@ ± 14 mA $_{max}$	100	190	Ω
I_{SN}	Secondary nominal r.m.s. current		25		mA
K_N	Conversion ratio		2500 : 1000		
V_C	Supply voltage (± 5 %)		± 12 .. 15		V
I_C	Current consumption		10 (@±15V)+I_s		mA
V_d	R.m.s. voltage for AC isolation test [1], 50 Hz, 1 mn		2.5		kV

Accuracy - Dynamic performance data

X_G	Overall Accuracy @ I_{PN} , T_A = 25°C	@ ± 12 .. 15 V	± 0.9		%
		@ ± 15 V (± 5 %)	± 0.8		%
ε_L	Linearity		< 0.2		%

			Typ	Max	
I_O	Offset current @ I_P = 0, T_A = 25°C			± 0.15	mA
I_{OT}	Thermal drift of I_O	0°C .. + 25°C	± 0.06	± 0.25	mA
		+ 25°C .. + 70°C	± 0.10	± 0.35	mA
t_r	Response time [2] @ 90 % of $V_{P max}$		40		μs

General data

T_A	Ambient operating temperature		0 .. + 70	°C
T_S	Ambient storage temperature		- 25 .. + 85	°C
R_P	Primary coil resistance @ T_A = 70°C		250	Ω
R_S	Secondary coil resistance @ T_A = 70°C		110	Ω
m	Mass		22	g
	Standards [3]		EN 50178	

Features

- Closed loop (compensated) voltage transducer using the Hall effect
- Insulated plastic case recognized according to UL 94-V0.

Principle of use

- For voltage measurements, a current proportional to the measured voltage must be passed through an external resistor **R** , which is selected by the user and installed in series with the primary circuit of the transducer.

Advantages

- Excellent accuracy
- Very good linearity
- Low thermal drift
- Low response time
- High bandwidth
- High immunity to external interference
- Low disturbance in common mode.

Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
- Uninterruptible Power Supplies (UPS)
- Power supplies for welding applications.

Notes : [1] Between primary and secondary
[2] R_1 = 25 kΩ (L/R constant, produced by the resistance and inductance of the primary circuit)
[3] A list of corresponding tests is available

981125/14

Dimensions LV 25-P (in mm. 1 mm = 0.0394 inch)

Bottom view

+HT   -HT

2 × ⌀ 0.635mm

3 × ⌀ 1mm

−   +   M

26
15.24
2 × 7.62

22.2
21.2

Back view

20.95   15.1   1.2

Right view

4 +/-0.3

3.44

20.32

5.44

16.45   4.5 +/-0.3

Top view

LEM® swiss made

CE

LV 25−P
00−00 00

M   +   −

Standard 00
or N° SP.

Year Week

Secondary terminals

Terminal + : supply voltage + 12 .. 15 V
Terminal M : measure
Terminal − : supply voltage − 12 .. 15 V

Connection

+HT

R1

+HT
LV 25 P   M
−HT

RM

+CV

−HT

Mechanical characteristics

- General tolerance   ± 0.2 mm
- Fastening & connection of primary   2 pins
                                                                0.635 x 0.635 mm
- Fastening & connection of secondary   3 pins ⌀ 1 mm
- Recommended PCB hole   1.2 mm

Remarks

- I_s is positive when V_p is applied on terminal +HT.
- This is a standard model. For different versions (supply voltages, turns ratios, unidirectional measurements...), please contact us.

Instructions for use of the voltage transducer model LV 25-P

Primary resistor R_1 : the transducer's optimum accuracy is obtained at the nominal primary current. As far as possible, R_1 should be calculated so that the nominal voltage to be measured corresponds to a primary current of 10 mA.

Example: Voltage to be measured V_{PN} = 250 V
a) R_1 = 25 kΩ/2.5 W, I_p = 10 mA   Accuracy = ± 0.8 % of V_{PN} (@ T_A = +25°C)
b) R_1 = 50 kΩ/1.25 W, I_p = 5 mA   Accuracy = ± 1.6 % of V_{PN} (@ T_A = +25°C)

Operating range (recommended) : taking into account the resistance of the primary windings (which must remain low compared to R_1 in order to keep thermal deviation as low as possible) and the isolation, this transducer is suitable for measuring nominal voltages from 10 to 500 V.

LEM reserves the right to carry out modifications on its transducers, in order to improve them, without previous notice.

Current Transducer LA 55-P

For the electronic measurement of currents : DC, AC, pulsed...,
with a galvanic isolation between the primary circuit (high power)
and the secondary circuit (electronic circuit).

I_{PN} = 50 A

5 YEARS WARRANTY CE
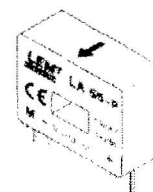
Electrical data

I_{PN}	Primary nominal r.m.s. current			50		A
I_P	Primary current, measuring range			0 .. ± 70		A
R_M	Measuring resistance @		T_A = 70°C		T_A = 85°C	
			$R_{M min}$ $R_{M max}$		$R_{M min}$ $R_{M max}$	
	with ± 12 V	@ ± 50 A $_{max}$	10 100		60 95	Ω
		@ ± 70 A $_{max}$	10 50		60[1] 60[1]	Ω
	with ± 15 V	@ ± 50 A $_{max}$	50 160		135 155	Ω
		@ ± 70 A $_{max}$	50 90		135[2] 135[2]	Ω
I_{SN}	Secondary nominal r.m.s. current			50		mA
K_N	Conversion ratio			1 : 1000		
V_C	Supply voltage (± 5 %)			± 12 .. 15		V
I_C	Current consumption			10 (@ ±15 V)+I_S		mA
V_d	R.m.s. voltage for AC isolation test, 50 Hz, 1 mn			2.5		kV

Accuracy - Dynamic performance data

X	Accuracy @ I_{PN} , T_A = 25°C	@ ± 15 V (± 5 %)		± 0.65	%
		@ ± 12 .. 15 V (± 5 %)		± 0.90	%
ε_L	Linearity			< 0.15	%
			Typ	Max	
I_O	Offset current @ I_P = 0, T_A = 25°C			± 0.2	mA
I_{OM}	Residual current [3] @ I_P = 0, after an overload of 3 x I_{PN}			± 0.3	mA
I_{OT}	Thermal drift of I_O	0°C .. + 70°C	± 0.1	± 0.5	mA
		- 25°C .. + 85°C	± 0.1	± 0.6	mA
t_{ra}	Reaction time @ 10 % of $I_{P max}$		< 500		ns
t_r	Response time @ 90 % of $I_{P max}$		< 1		µs
di/dt	di/dt accurately followed		> 200		A/µs
f	Frequency bandwidth (- 1 dB)		DC .. 200		kHz

General data

T_A	Ambient operating temperature		- 25 .. + 85	°C
T_S	Ambient storage temperature		- 40 .. + 90	°C
R_S	Secondary coil resistance @	T_A = 70°C	80	Ω
		T_A = 85°C	85	Ω
m	Mass		18	g
	Standards [4]		EN 50178	

Notes : [1] Measuring range limited to ± 60 A $_{max}$
[2] Measuring range limited to ± 55 A $_{max}$
[3] Result of the coercive field of the magnetic circuit
[4] A list of corresponding tests is available

Features

- Closed loop (compensated) current transducer using the Hall effect
- Printed circuit board mounting
- Insulated plastic case recognized according to UL 94-V0.

Advantages

- Excellent accuracy
- Very good linearity
- Low temperature drift
- Optimized response time
- Wide frequency bandwidth
- No insertion losses
- High immunity to external interference
- Current overload capability.

Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
- Uninterruptible Power Supplies (UPS)
- Switched Mode Power Supplies (SMPS)
- Power supplies for welding applications.

980706/8

Dimensions LA 55-P (in mm. 1 mm = 0.0394 inch)

Bottom view

14.45

31.5

Left view

7.9

6.55

27.6

4.5 +/-.03

12.7

LEM® LA 55-P

CE   swiss
made

M — 00-00 00   +

11.8

7.9   27.2

0.4

5.08   22.86

36.5

Standard 00
or N° SP..

Year Week

Front view

Secondary terminals

Terminal + : supply voltage + 12 .. 15 V
Terminal - : supply voltage - 12 .. 15 V
Terminal M : measure

Connection

LA 55-P

+ ──────── o +
- ──────── o -

M ── I_s ──⊘── R_M ── o 0V
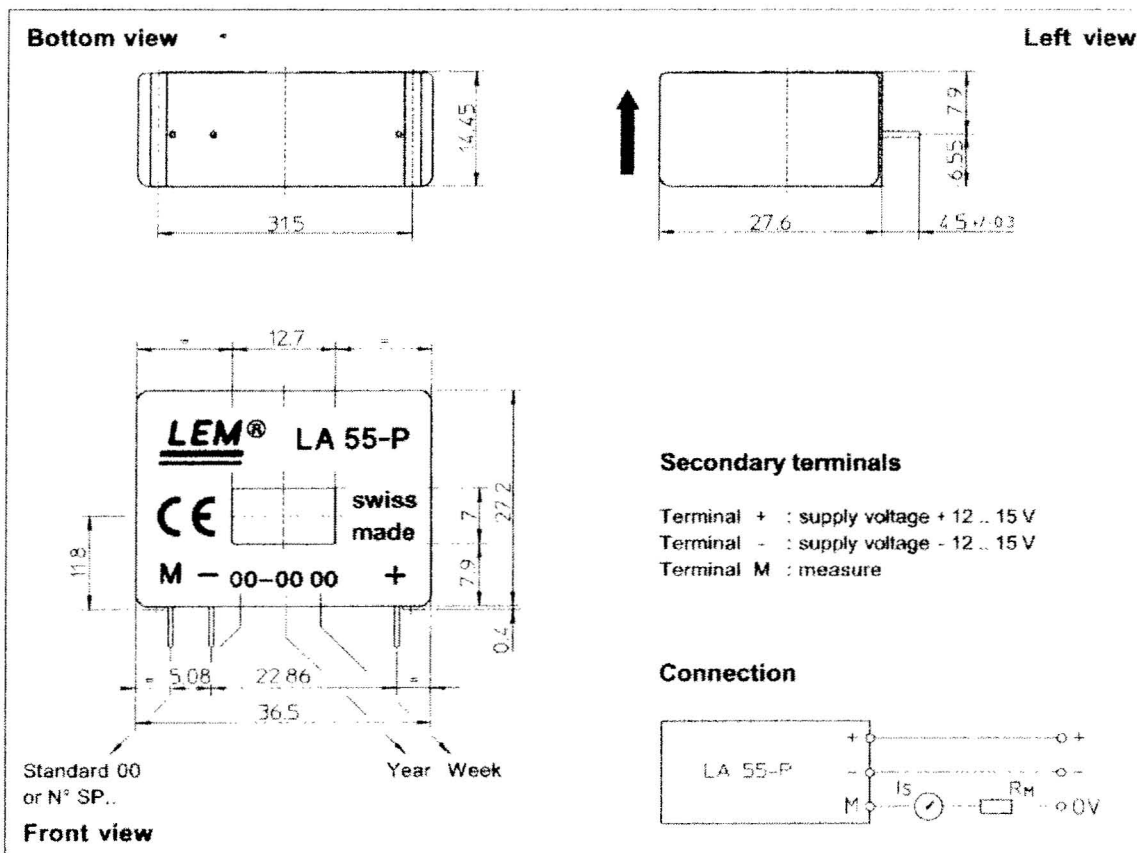
Mechanical characteristics

- General tolerance   ± 0.2 mm
- Primary through-hole   12.7 x 7 mm
- Fastening & connection of secondary   3 pins
                                        0.63 x 0.56mm
  Recommended PCB hole   0.9 mm

Remarks

- I_s is positive when I_p flows in the direction of the arrow.
- Temperature of the primary conductor should not exceed 90°C.
- Dynamic performances (di/dt and response time) are best with a single bar completely filling the primary hole.
- In order to achieve the best magnetic coupling, the primary windings have to be wound over the top edge of the device.
- This is a standard model. For different versions (supply voltages, turns ratios, unidirectional measurements...), please contact us.

LEM reserves the right to carry out modifications on its transducers, in order to improve them, without previous notice.

Current Transducer LA 205-S

For the electronic measurement of currents : DC, AC, pulsed...,
with a galvanic isolation between the primary circuit (high power)
and the secondary circuit (electronic circuit).

I_{PN} = 200 A

CE

Electrical data

				$T_A = 70°C$		$T_A = 85°C$		
I_{PN}	Primary nominal r.m.s. current		200					A
I_p	Primary current, measuring range		0 .. ± 300					A
$I_{p\,max}$	Measuring overload [1]		600					A
R_M	Measuring resistance @			$R_{M\,min}$	$R_{M\,max}$	$R_{M\,min}$	$R_{M\,max}$	
	with ± 12 V	@ ± 200 A $_{max}$		0	68	0	66	Ω
		@ ± 300 A $_{max}$		0	33	0	30	Ω
	with ± 15 V	@ ± 200 A $_{max}$		5	95	5	93	Ω
		@ ± 300 A $_{max}$		5	50	5	49	Ω
I_{SN}	Secondary nominal r.m.s. current		100					mA
K_N	Conversion ratio		1 : 2000					
V_c	Supply voltage (± 5 %)		± 12 .. 15					V
I_c	Current consumption		20 (@ ± 15V) + I_s					mA
V_b	R.m.s rated voltage [2], safe separation		1625					V
	basic isolation		3250					V

Accuracy - Dynamic performance data

				Typ	Max	
X_G	Overall accuracy @ I_{PN}, $T_A = 25°C$		± 0.8			%
ε_L	Linearity		< 0.1			%
I_o	Offset current @ $I_p = 0$, $T_A = 25°C$				± 0.15	mA
I_{OM}	Residual current [3] @ $I_p = 0$, after an overload of 3 x I_{PN}				± 0.50	mA
I_{OT}	Thermal drift of I_o	- 10°C .. + 85°C		± 0.15	± 0.30	mA
t_{ra}	Reaction time @ 10 % of $I_{p\,max}$		< 500			ns
t_r	Response time [4] @ 90 % of $I_{p\,max}$		< 1			µs
di/dt	di/dt accurately followed		> 100			A/µs
f	Frequency bandwidth (- 3 dB)		DC .. 100			kHz

General data

T_A	Ambient operating temperature		- 10 .. + 85	°C
T_s	Ambient storage temperature		- 40 .. + 90	°C
R_s	Secondary coil resistance @	$T_A = 70°C$	35	Ω
		$T_A = 85°C$	37	Ω
m	Mass		110	g
	Standards [5]		EN 50178	

Notes : [1] 3 mn/hour @ $V_c = ± 15$ V, $R_M = 5$ Ω
[2] Pollution class nr 2. With a non insulated primary bar which fills the
through-hole
[3] The result of the coercive field of the magnetic circuit
[4] With a di/dt of 100 A/µs
[5] A list of corresponding tests is available

Features

- Closed loop (compensated) current transducer using the Hall effect
- Insulated plastic case recognized according to UL 94-V0
- Patent pending.

Advantages

- Excellent accuracy
- Very good linearity
- Low temperature drift
- Optimized response time
- Wide frequency bandwidth
- No insertion losses
- High immunity to external interference
- Current overload capability

Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
- Uninterruptible Power Supplies (UPS)
- Switched Mode Power Supplies (SMPS)
- Power supplies for welding applications.

980716/5

Dimensions LA 205-S (in mm. 1 mm = 0.0394 inch)
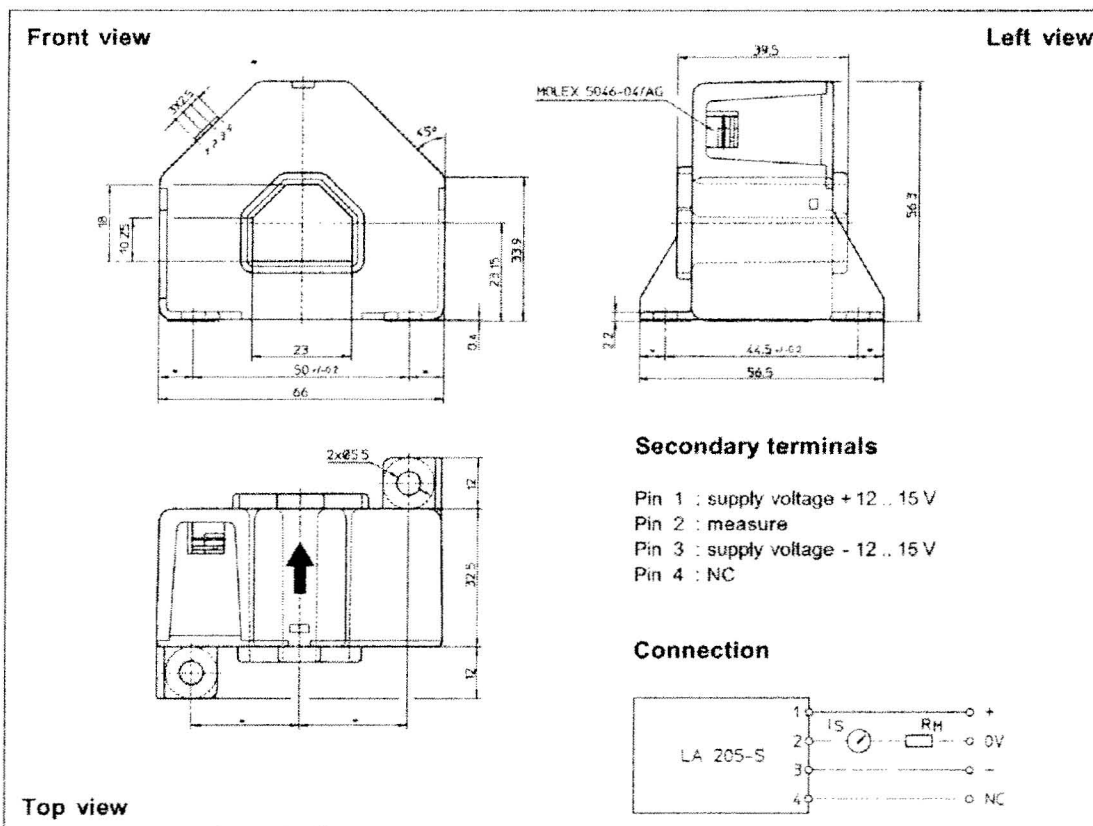
Front view

Left view

MOLEX 5046-04/AG

Top view

Secondary terminals

Pin 1 : supply voltage + 12 .. 15 V
Pin 2 : measure
Pin 3 : supply voltage - 12 .. 15 V
Pin 4 : NC

Connection

LA 205-S

Mechanical characteristics

- General tolerance   ± 0.5 mm
- Fastening   2 holes ⌀ 5.5 mm
- Primary through-hole   23 x 18 mm
- Connection of secondary Molex 5046-04/AG

Remarks

- I_s is positive when I_p flows in the direction of the arrow.
- Temperature of the primary conductor should not exceed 100°C.
- Dynamic performances (di/dt and response time) are best with a single bar completely filling the primary hole.
- This is a standard model. For different versions (supply voltages, turns ratios, unidirectional measurements...), please contact us.

Current Transducer LA 305-S

For the electronic measurement of currents : DC, AC, pulsed...,
with a galvanic isolation between the primary circuit (high power)
and the secondary circuit (electronic circuit).

I_{PN} = 300 A

CE

Electrical data

I_{PN}	Primary nominal r.m.s. current			300		A
I_P	Primary current, measuring range			0 .. ± 500		A
R_M	Measuring resistance @		$T_A = 70°C$		$T_A = 85°C$	
			$R_{M min}$	$R_{M max}$	$R_{M min}$	$R_{M max}$	
	with ± 12 V	@ ± 300 A $_{max}$	0	52	0	50	Ω
		@ ± 500 A $_{max}$	0	17	0	15	Ω
	with ± 15 V	@ ± 300 A $_{max}$	0	75	5	73	Ω
		@ ± 500 A $_{max}$	0	31	5	29	Ω
I_{SN}	Secondary nominal r.m.s. current			120		mA
K_N	Conversion ratio			1 : 2500		
V_C	Supply voltage (± 5 %)			± 12 .. 15		V
I_C	Current consumption			20 (@ ± 15 V) + I_S		mA
V_D	R.m.s. rated voltage [1], safe separation			1750		V
	basic isolation			3500		V

Accuracy - Dynamic performance data

			Typ	Max	
X_G	Overall accuracy @ I_{PN}, $T_A = 25°C$			± 0.8	%
$ε_L$	Linearity			< 0.1	%
I_O	Offset current @ $I_P = 0$, $T_A = 25°C$			± 0.20	mA
I_{OM}	Residual current [2] @ $I_P = 0$, after an overload of 3 x I_{PN}			± 0.40	mA
I_{OT}	Thermal drift of I_O	- 10°C .. + 85°C	± 0.12	± 0.30	mA
t_{ra}	Reaction time @ 10 % of $I_{P max}$			< 500	ns
t_r	Response time [3] @ 90 % of $I_{P max}$			< 1	µs
di/dt	di/dt accurately followed			> 100	A/µs
f	Frequency bandwidth (- 3 dB)			DC .. 100	kHz

General data

T_A	Ambient operating temperature		- 10 .. + 85	°C
T_S	Ambient storage temperature		- 40 .. + 90	°C
R_S	Secondary coil resistance @	$T_A = 70°C$	35	Ω
		$T_A = 85°C$	37	Ω
m	Mass		200	g
	Standards [4]		EN 50178	

Features

- Closed loop (compensated) current transducer using the Hall effect
- Insulated plastic case recognized according to UL 94-V0
- Copyright protected.

Advantages

- Excellent accuracy
- Very good linearity
- Low temperature drift
- Optimized response time
- Wide frequency bandwidth
- No insertion losses
- High immunity to external interference
- Current overload capability.

Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
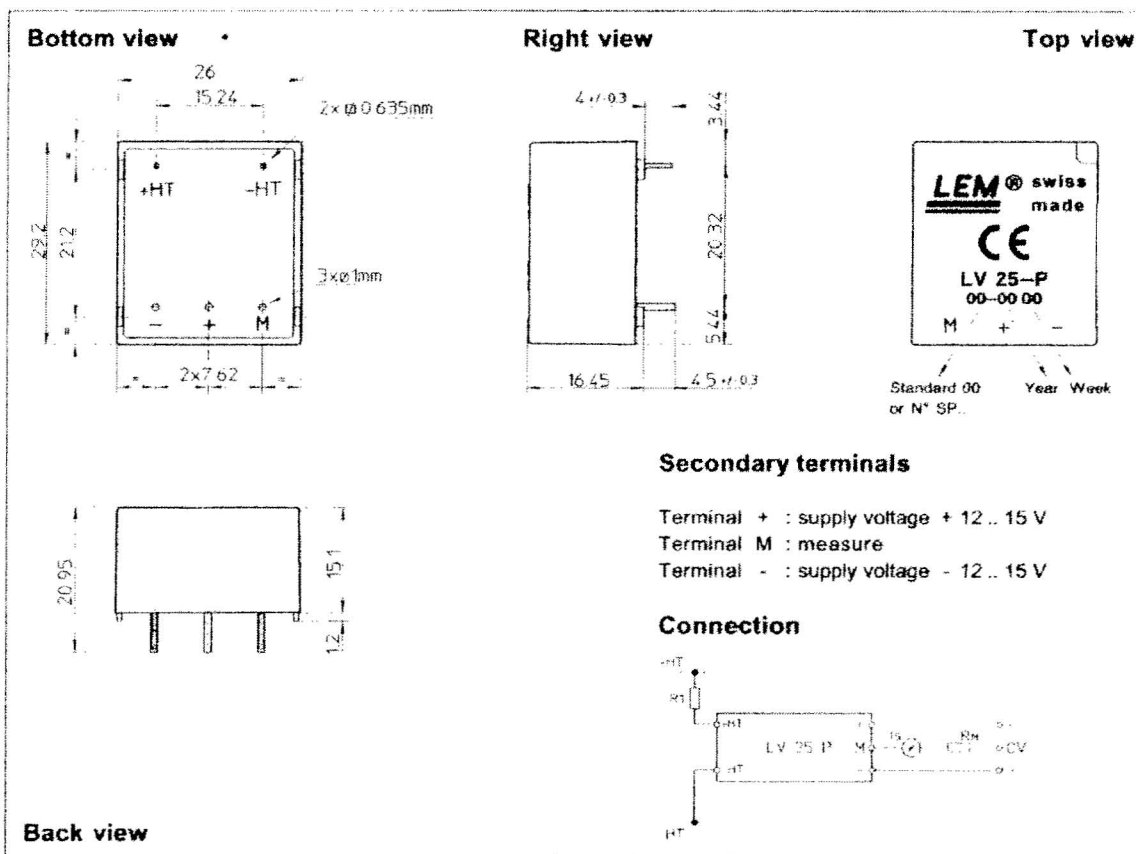- Uninterruptible Power Supplies (UPS)
- Switched Mode Power Supplies (SMPS)
- Power supplies for welding applications.

Notes : [1] Pollution class 2. With a non insulated primary bar which fills the through-hole
[2] The result of the coercive field of the magnetic circuit
[3] With a di/dt of 100 A/µs
[4] A list of corresponding tests is available

980707/6

Dimensions LA 305-S (in mm. 1 mm = 0.0394 inch)
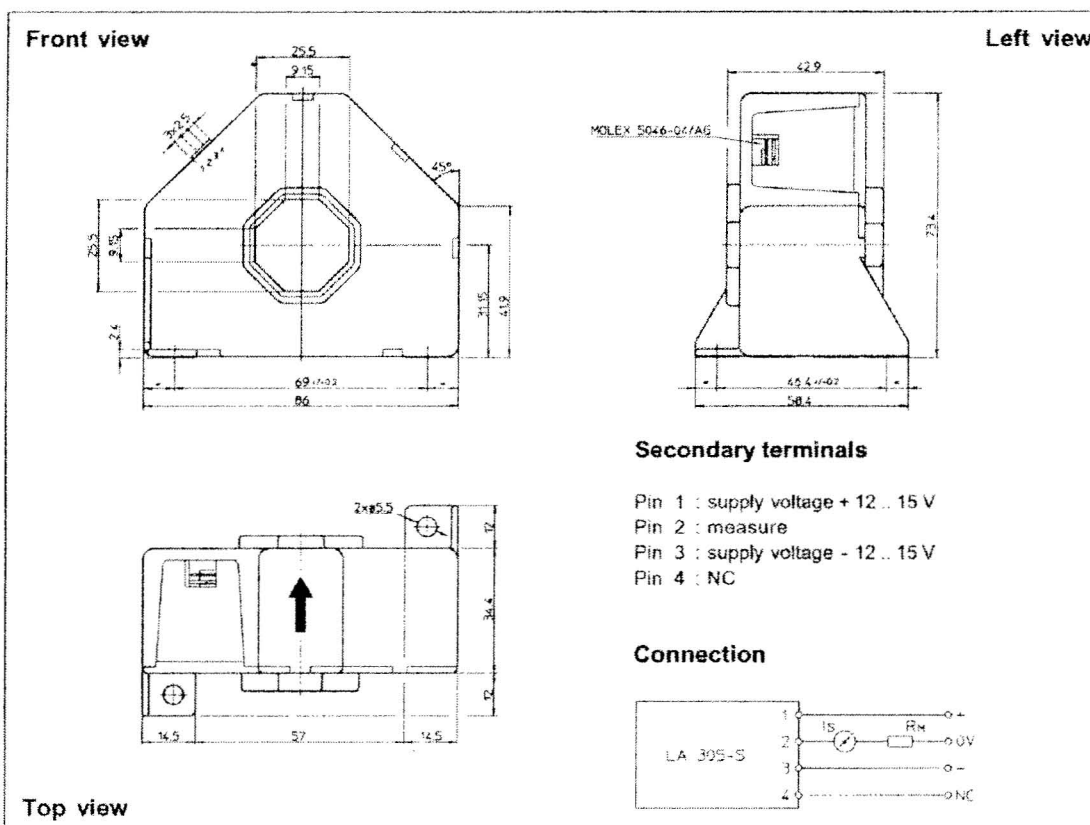
Front view

Left view

MOLEX 5046-04/AG

Secondary terminals

Pin 1 : supply voltage + 12 .. 15 V
Pin 2 : measure
Pin 3 : supply voltage - 12 .. 15 V
Pin 4 : NC

Connection

LA 305-S

Top view

Mechanical characteristics

- General tolerance   ± 0.5 mm
- Fastening   2 holes ⌀ 5.5 mm
- Primary through-hole   25.5 x 25.5 mm
- Connection of secondary   Molex 5046-04/AG

Remarks

- I_s is positive when I_p flows in the direction of the arrow.
- Temperature of the primary conductor should not exceed 100°C.
- Dynamic performances (di/dt and response time) are best with a single bar completely filling the primary hole.
- This is a standard model. For different versions (supply voltages, turns ratios, unidirectional measurements ..), please contact us.

FLUKE.

i3000s

AC Current Clamp

Instruction Sheet

Introducing the i3000s

The i3000s is a clamp-on AC Current Clamp for use with oscilloscopes and multimeters with banana input using the supplied Dual Banana to BNC Adapter. The clamp is optimized for measurement on power distribution systems.

Unpacking

The following items should be included in your Current Clamp box.
- Current Clamp
- Dual Banana to BNC Adapter (model PM9081/001)
- Instruction Sheet (this paper)

Check the contents of the shipping box for completeness. If something in the box has been damaged or missing, contact your distributor or the nearest FLUKE sales or service office immediately.

Safety Information

⚠️⚠️ **Read First: Safety Information.**

To ensure safe operation and service of the current clamp, follow these instructions:

- Read the operating instructions before use and follow all safety instructions.
- Use the Current Clamp only as specified in the operating instructions, otherwise the clamp's safety features may not protect you.
- Adhere to local and national safety codes. Individual protective equipment must be used to prevent shock and arc blast injury where hazardous live conductors are exposed.
- Do not hold the Current Clamp anywhere beyond the tactile barrier, see Figure 8.
- Before each use, inspect the Current Clamp. Look for cracks or missing portions of the clamp housing or output cable insulation. Also look for loose or weakened components. Pay particular attention to the insulation surrounding the jaws.
- Check the magnetic mating surfaces of the clamp jaws; these should be free of dust, dirt, rust and other foreign matter.
- Never use the clamp on a circuit with voltages higher than 600 V CAT III.
  - CAT III equipment is designed to protect against transients in equipment in fixed equipment installations, such as distribution panels, feeders and short branch circuits, and lighting systems in large buildings.

- Use extreme caution when working around bare conductors or bus bars. Contact with the conductor could result in electric shock.
- Use caution when working with voltages above 60 V dc, 30 V ac rms or 42 V ac peak. Such voltages pose a shock hazard.

Symbols

Symbol	Description
⚡	May be used on HAZARDOUS LIVE conductors.
▣	Product is protected by double insulation.
⚠️	Risk of Danger, important information. See Instruction Sheet.
⚠️	Risk of Electric Shock
CE	Conforms to relevant European standards
⏚	Earth ground

Specifications

SAFETY

⚠️	Input jaws & Output float voltage	Complies with European standards EN/IEC 61010-1 2nd Edition & EN/IEC 61010-02-032 for 600V CAT III, pollution degree 2.
	CE Conformity	EN/IEC 61010-02-032

ELECTRICAL SPECIFICATIONS

All Electrical Specifications are valid at the following reference conditions:
- *Ambient temperature* 23±3°C (73±3°F)
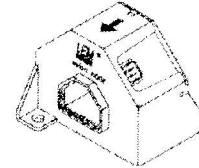- *Relative Humidity* 0 to 85%
- *Frequency* 48 to 65 Hz
- *Continuous external field* < 40 A/m
- *Load impedance* 1 MΩ // 47 pF
- *The current may not contain any DC component*
- *No influence from adjacent currents*
- *The conductor must be centered within the jaw aperture*

		Current Ranges	
	30A	300A	3000A
Measuring range	1 to 30A ac	1 to 300A ac	1 to 3000 A ac
Crest factor *	up to 3	up to 3	up to 3
Maximum non-destructive current:			
· continuous	35A	350A	2400A
· 30min.+30min. stop			2800A
· 5min.+30min. stop			4000A
Output signal	10 mV/A	1 mV/A	0.1 mV/A
Basic accuracy	2% of reading + 0.1A	2% of reading + 0.5A	2% of reading + 2A
Additional errors:			
· with temperature (-10 to +50°C)	0.02%/°C	0.02%/°C	0.02%/°C
· with position of conductor in the clamp window	1% + 0.1A	1% + 0.1A	1% + 0.1A
· with adjacent conductor	0.005A/A	0.005A/A	0.005A/A
· with frequency (10Hz to 100 kHz)	See Figure 1	See Figure 2	See Figure 3
Phase shift	20° @ 2A 10° @ 6A 5° @ 30A See Figure 4	3° @ 20A 1.5° @ 60A 1° @ 300A See Figure 5	3° @ 200A 1.5° @ 600A 1° @ 3000A See Figure 6

* This is the maximum permissible ratio between the peak value of a superimposed transient and the ac rms value.

Bandwidth (-3dB)   10 Hz to 100 kHz
(see Figure 7 for derating)

GENERAL

Clamp Dimensions	310 × 120 × 46 mm (12.2 × 4.7 × 1.9 in)
Weight	1200 g (42.3 oz)
Cable length	2.1 m (82.6 in)
Maximum conductor size	⌀ 64 mm (2.5 in)
Maximum Jaw Opening	90 mm (3.5 in)
Temperature	
Operating	-10 to +50°C (+14 to +122°F)
Non-operating	-40 to +85°C (-40 to +185°F)
Relative Humidity	
Operating	0 to 85%, up to +35°C (+95°F) 0 to 75%, up to +50°C (+122°F)
Non-operating	0 to 85%, up to -10°C (+14°F) 0 to 75%, up to +85°C (+185°F)
Altitude	
Operating	to 2000 m (6500 ft)
Non-operating	to 12000 m (40000 ft)

FREQUENCY RESPONSE

30A Range
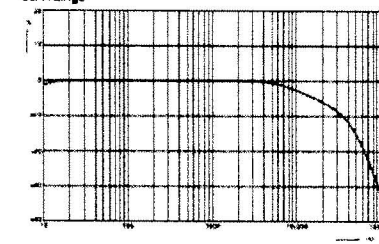
Figure 1. Frequency Response @ 10A

300A Range
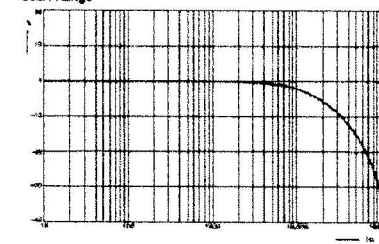
Figure 2. Frequency Response @ 10A

3000A Range
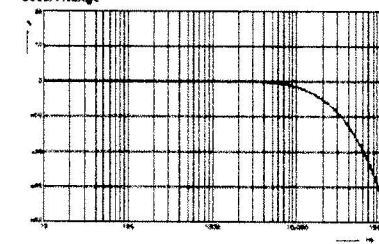
Figure 3. Frequency Response @ 100A

4822872008839

PHASE SHIFT

30A Range
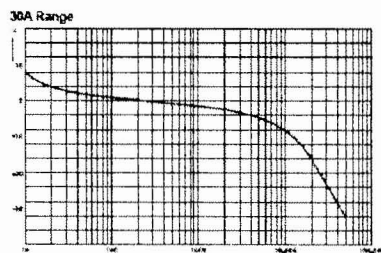
Figure 4. Phase Shift @ 10A

300A Range
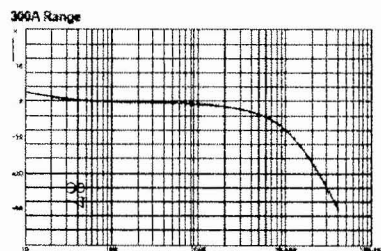
Figure 5. Phase Shift @ 10A

3000A Range
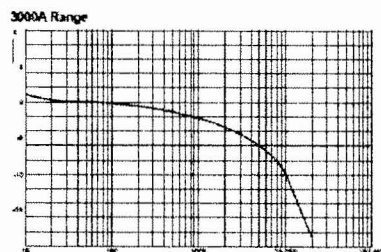
Figure 6. Phase Shift @ 100A
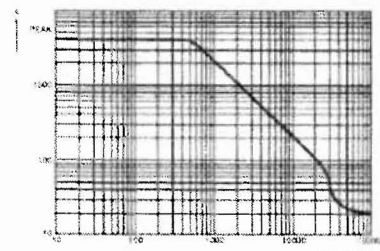
Figure 7. Maximum Current as a Function of the Frequency

Instrument Compatibility

The i3000s is compatible with any Fluke ScopeMeter test tool, Power Harmonics Analyzer, Oscilloscope, Multimeter, or other voltage measurement device that has the following features.

- BNC input connector. The Dual Banana to BNC Adapter included in the package, can be used to connect to standard inputs on multimeters. For the 120 series ScopeMeters use the BB120 Shielded Banana to BNC Adapter.
- Input accuracy of 2% or better to take full advantage of the accuracy of the Current Clamp.
- Input impedance of greater than or equal to 1 MΩ, and for full bandwidth and accuracy a maximum input capacity of 47 pF.
- A passband of more than four times the frequency of the waveform to be measured.

Using the Current Clamp

To use the Current Clamp, follow these instructions:

1. Connect the i3000s Current Clamp to the desired input on the measuring instrument. When you are using a multimeter, use the Dual Banana to BNC Adapter (PM9081/001) to connect the Current Clamp to the input. (See Figure 8.)
2. On the Current Clamp, select the least sensitive range (0.1 mV/A).
3. Select the appropriate clamp sensitivity on your ScopeMeter test tool or oscilloscope.
4. Position the Current Clamp perpendicularly and centered around the conductor.
   Make sure that the arrow marked on the clamp jaw points toward the correct orientation for correct phase display on the oscilloscope. (See Figure 9.)
   Use the markings on the jaws to center the conductor.
5. Observe the current value and waveform on the instrument's display.
6. If desired, select a lower range on the Current Clamp and set the corresponding sensitivity (mV/A setting) on the ScopeMeter test tool or oscilloscope.

If the corresponding sensitivity is not available on the ScopeMeter or oscilloscope, select the closest setting and calculate the actual current value from the displayed.

Example 1: Current Clamp set to 0.1 mV/A, ScopeMeter test tool set to 1 mV/A.
The ScopeMeter test tool displays 50A.
The real current value can be calculated with the following formula:

Actual current =

$$\text{display value} \times \frac{\text{sensitivity instrument}}{\text{sensitivity Current Clamp}} = 50 \times \frac{1mV / A}{0.1mV / A} = 500 A$$

Example 2: Current Clamp set to 10 mV/A. Multimeter displays 0.285V.

Actual current =

$$\frac{\text{display value}}{\text{sensitivity Current Clamp}} = \frac{0.285V}{10mV / A} = \frac{285 mV}{10mV / A} = 28.5 A$$
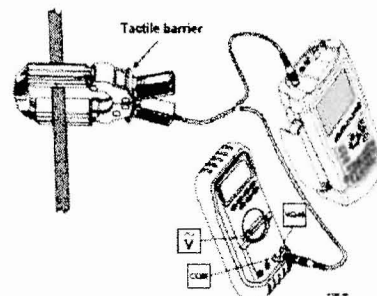
Figure 8. Measurement Setup

⚠ Warning

If the sensitivity setting (mV/A) of the ScopeMeter test tool or oscilloscope does not correspond with the setting of the Current Clamp, the instrument may display a much lower current than the actual value. This may cause misinterpretation and as a consequence incorrect handling.
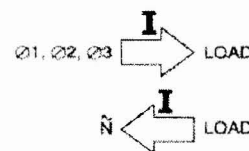
Figure 9. Orientation of the Current Clamp

Measurement Considerations

Observe the following guidelines for positioning the Current Clamp Jaws:

- Center the conductor inside the clamp jaws.
- Make sure the clamp is perpendicular to the conductor.
- Make sure that the arrow marked on the jaw of the Current Clamp points toward the correct direction.

Observe the following guidelines when making measurements:

- If possible, avoid measurements close to other current-carrying conductors.
- On the Current Clamp, select the most appropriate range for the current being measured to get the best accuracy.

Maintenance

Before each use, assure continued safety by inspecting the clamp. Look for cracks or missing portions of the clamp housing and output cable insulating cover and for loose or weakened components. Pay particular attention to the insulation surrounding the clamp jaws. If a clamp fails this inspection, tape it shut to prevent unintended operation. A clamp under warranty will be promptly repaired or replaced (at Fluke's discretion) and returned at no charge.

Cleaning and Storage

Periodically wipe the case with a damp cloth and detergent; do not use abrasives or solvents. Open the jaws and wipe the magnetic pole pieces with a lightly oiled cloth. Do not allow rust or corrosion to form on the magnetic core ends.

If your Current Clamp does not work

If the Current Clamp does not perform properly, use the following steps to help isolate the problem:

- Inspect the jaw mating surface for cleanliness. If any foreign material is present, the jaws will not close properly and errors will result.
- Verify that the function selection on the ScopeMeter test tool or oscilloscope is correct, i.e., the display vertical resolution is not too low or too high.

LIMITED WARRANTY & LIMITATION OF LIABILITY

This Fluke product will be free from defects in material and workmanship for one year from the date of purchase. This warranty does not cover fuses, disposable batteries or damage from accident, neglect, misuse or abnormal conditions of operation or handling. Resellers are not authorized to extend any other warranty on Fluke's behalf. To obtain service during the warranty period, send your defective product to the nearest Fluke Authorized Service Center with a description of the problem.

THIS WARRANTY IS YOUR ONLY REMEDY. NO OTHER WARRANTIES, SUCH AS FITNESS FOR A PARTICULAR PURPOSE, ARE EXPRESSED OR IMPLIED. FLUKE IS NOT LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR LOSSES ARISING FROM ANY CAUSE OR THEORY.

Since some states or countries do not allow the exclusion or limitation of an implied warranty or of incidental or consequential damages, this limitation of liability may not apply to you.

Fluke Corporation
P.O. Box 9090
Everett, WA
98206-9090, USA

Fluke Industrial B.V.
P.O. Box 90
7600 AB Almelo
The Netherlands

SERVICE CENTERS

To locate an authorized service center, visit us on the World Wide Web

http://www.fluke.com

or call Fluke using any of the phone numbers listed below:

+1-888-993-5853 in U.S.A. and Canada

+31-40-267-5200 in Europe

+1-425-446-5500 from other countries

RUGGED SOLID STATE TRANSDUCERS
WITH AMPLIFIED OUTPUTS
GAGE, ABSOLUTE, AND COMPOUND PRESSURES

1 NEW

EXCLUSIVE!

PX209/PX219 Series
0-15 to 0-300 psi
0-1 to 0-20 bar

Starts at
$195

Based on proprietary solid state sensor technology originally used in military aviation applications.

Connector included.

PX209-100G5V, cable style, $195, shown smaller than actual size.

PX219-100G5V, connector style, $195, shown actual size.

MADE IN USA · NIST · CE

- ✓ Stainless Steel Fitting and Body
- ✓ 5-Point NIST-Traceable Calibration Included
- ✓ Solid State Media Isolation (Suitable for Use with Most Industrial Fluids and Oils)
- ✓ Broad Temperature-Compensated Range of -20 to 80°C (-4 to 176°F) Yields High Stability with Changing Temperatures
- ✓ Electrical Isolation to 100 MΩ Ensures Long-Term Reliability
- ✓ Rugged High Shock and Vibration Design for Tough OEM Applications
- ✓ 100,000 Hr MTBF Typical

Based on proprietary sensor technology developed by OMEGA to meet the high reliability and accuracy demanded by military applications, the PX209/219 Series voltage and current output pressure transducer offers superior performance in non-corrosive applications, including: engine/powertrain testing, well monitoring, jet fuel pressure metering, and ground and race water monitoring. The transducer uses a 4-active-arm bridge sensor with a micro-machined diffused silicon diaphragm and proprietary thin-film media, plus dielectric isolation barriers.

Current Output			Voltage Output		
PIN		WIRE	PIN		WIRE
1	+EXC	RED	1	+EXC	RED
2	-EXC	BK	2	COM	BK
			3	-OUT	WT

14.2 (0.56) 59.2 (2.33) 34.4 (1.35)

19 (0.75) 25.4 (1.00) 27.2 (1.07)

Dimensions: mm (in)
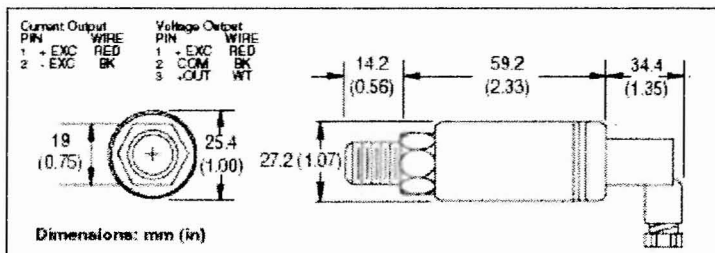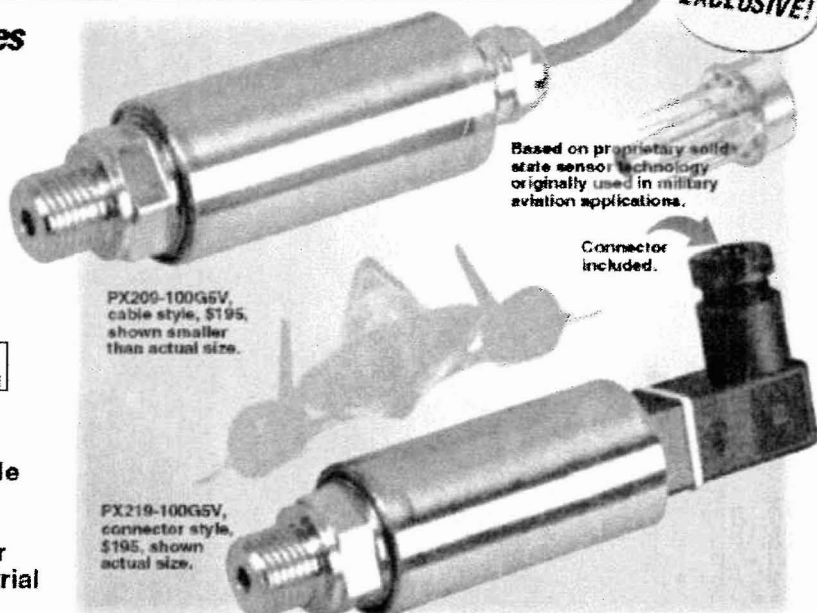
This same core sensing element technology, which includes multiple types of signal conditioning and the ability to survive extremes of shock and vibration, provides a modular building block for OMEGA's revolutionary family of pressure-sensing instruments.

SPECIFICATIONS
Voltage Output
Excitation: 24 Vdc @ 15 mA
  5 Vdc Output: 7 to 35 Vdc
  10 Vdc Output: 12 to 35 Vdc
Output: 0 to 5 Vdc or 0 to 10 Vdc, ±1.5% FSO, 3-wire
Zero Balance: 0 Vdc ±2% FSO

4 to 20 mA Output
Excitation: 24 Vdc (7 to 35 Vdc) reverse polarity protected
  Output: 4 to 20 mA (2-wire) ±1% FSO
  Zero Balance: 4 mA ±2% FSO
Max Loop Resistance:
50 x (supply voltage - 10) Ω

Common Specifications
Accuracy: 0.25% FS (including linearity, hysteresis and repeatability)
Operating Temperature:
-54 to 121°C (-65 to 250°F)
Compensated Temperature:
-20 to 80°C (-4 to 176°F)
Thermal Effects:
0.04% FS/°C (0.02% FS /°F)
Proof Pressure: 150%
Burst Pressure: 300% range max
Response Time: 2 ms typical

B-72

88

Appendix B

LPAC System Diagram

LP SERVICE AIR SYSTEM
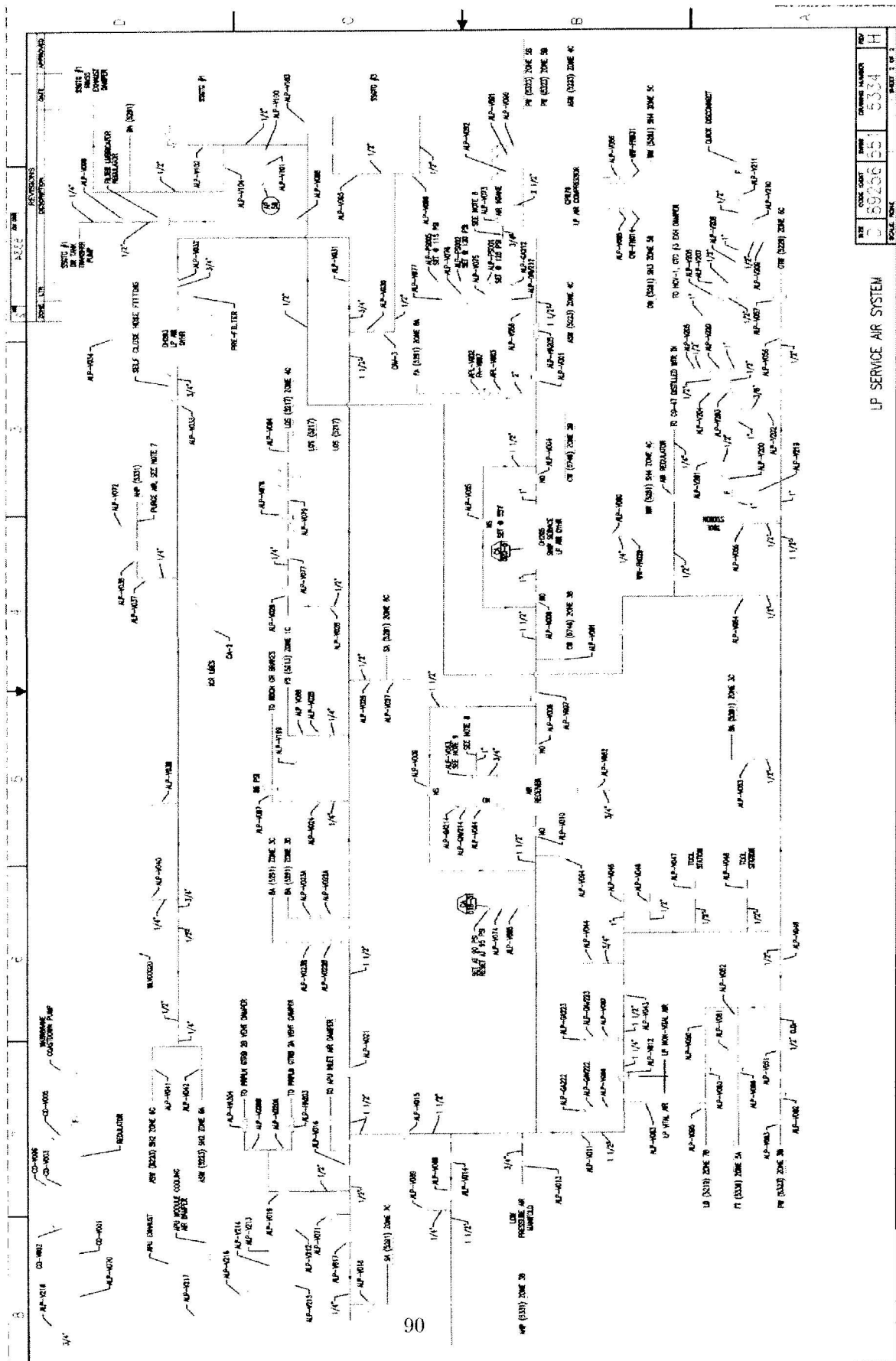
Appendix C

LPAC Test Procedure

LBES Non-Intrusive Load Monitor LPAC Testing

C.1   Objective

Observe the operation of the Low Presure Air Compressor (LPAC) using the Non-Intrusive Load Monitor (NILM) to attempt to determine the cause of abnormal operation detected during previous operations.

C.2   Procedure

1. Verify the NILM for LPAC #1 is operational.

2. Verify aggragate sensor NILMs are operational.

3. Verify the Low Pressure Air System is lined up to support operation of the LPAC #1.

4. Start LPAC #1 in automatic mode.

5. Insert a 12.5 SCFM into the LP Air System using the flow meter installed on ALP-V033.

6. Observe operation of the LPAC through a number of load and unload cycles.

7. Isolate shop air by shutting FA-V007.

8. Observe operation of the LPAC through a number of load and unload cycles.

9. Open FA-V007.

10. Secure LPAC #1

11. Verify the NILM for LPAC #2 is operational.

12. Verify the Low Pressure Air System is lined up to support operation of the LPAC #2.

13. Start LPAC #2 in automatic mode.

14. Observe operation of the LPAC through a number of load and unload cycles.

15. Isolate shop air by shutting FA-V007.

16. Observe operation of the LPAC through a number of load and unload cycles.

17. Open FA-V007.

18. Secure LPAC #2.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix D

Amplifier for 1SA & 3SA NILM

This amplifier was designed by Professor Cox to amplify the voltage signals from the Fluke i3000S. The schematic diagram for the circuits are shown in Figure D-1. The design allows the scale resistors to be changed out easily, allowing the amplication factors to be changed as required. The OpAmps can also be changed easily to any 8-pin OpAmp desired. The power terminals of the amplifer are connected the corresponding terminals on the NILM power connection block. The output terminals of the amplifier are connected to the current positions of SCSI terminal board inside of the NILM enclosure. Unlike a normal NILM setup, reference resistors are not required for the current measurements because the Fluke i3000S CTs output a voltage directly. The voltage terminals of the SCSI terminal board are connected to the output terminals of the voltage sensing board. Figure D-2 shows the design of the amplifier board in ExpressPCB^{TM}software.
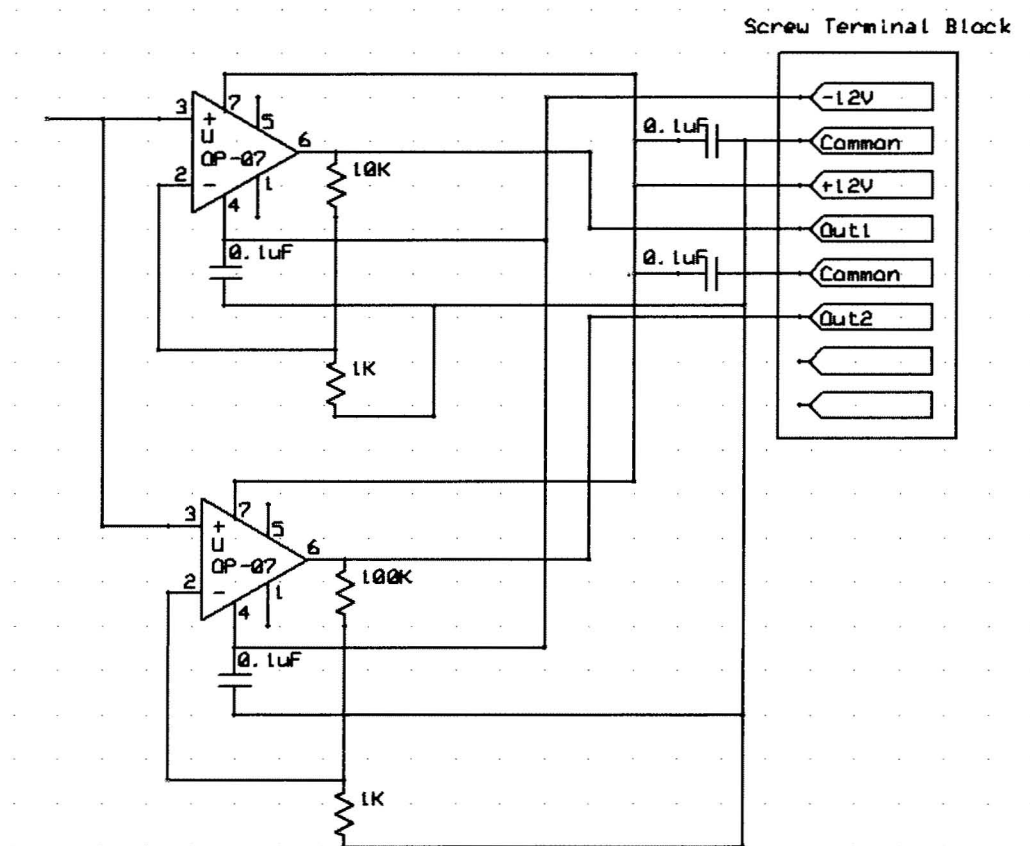
Figure D-1: Schematic Diagram of the Amplifier for the 1SA & 3SA NILM Systems usinig Fluke i3000S CTs.
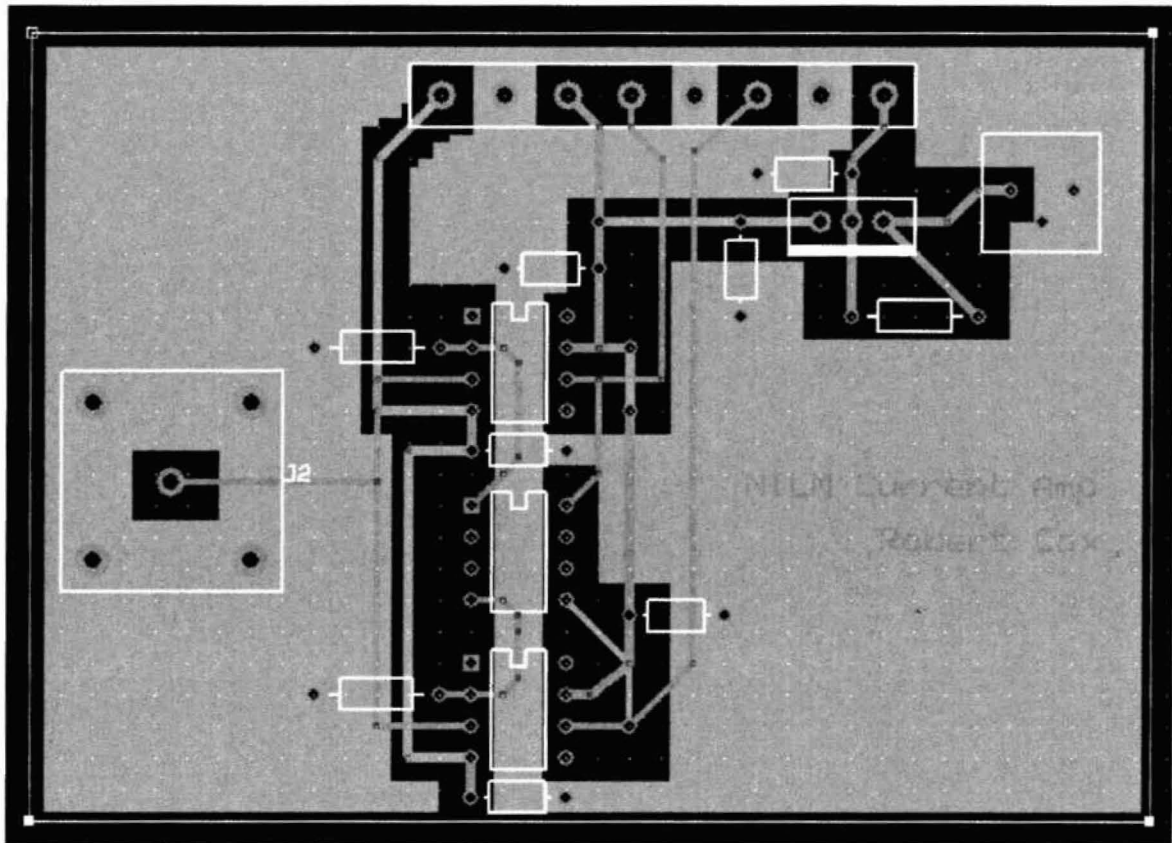
Figure D-2: ExpressPCB^{TM}Image of the Amplifier for the 1SA & 3SA NILM Systems usinig Fluke i3000S CTs.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix E

Processing Scripts for NILM Data

These scripts were written by previous researchers including LCDR Bennett [3], Professor Rob Cox [7] and Jim Paris [?]. They have been modified as required during this research to account for new data sources and local data storage and processing.

E.1 Reading Data DVDs

E.1.1 1SA

```
#!/bin/bash
for i in *.gz
do
echo $i;
gunzip -c $i > tempfile;
grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 | prep > 1SA_fullprep${i/.gz/.txt}
cat 1SA_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > 1SA${i/.gz/.txt}
rm -f $i
done
rm -f tempfile
```

E.1.2 3SA

```
#!/bin/bash
for i in *.gz
do
echo $i;
gunzip -c $i > tempfile;
grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 | prep > 3SA_fullprep${i/.gz/.txt}
cat 3SA_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > 3SA${i/.gz/.txt}
rm -f $i
done
rm -f tempfile
```

E.1.3 LPAC #2

The script for LPAC #2 requies different processing because the data for the pressure transducer is included in the data file.

```
#!/bin/bash

#mount /cdrom
#cp /cdrom/* .

for i in *.1.gz
do
echo $i;
gunzip -c $i > tempfile;
grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 | prep > LPAC2_fullprep${i/.gz/.txt}
cat LPAC2_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 1 2 > LPAC2${i/.gz/.txt}
rm -f $i
done
for i in *.2.gz
do
echo $i;
gunzip -c $i > tempfile;
grep -v ^# tempfile | ./chansift2_12bit.pl 1 > LPAC2_pressure_${i/.gz/.txt}
rm -f $i
done

rm -f tempfile
#rm -f LPAC*

#umount /cdrom
#eject
```

The \textit{Prune Script} reduces the size of the pressure files by a factor specified by t

```bash
#!/bin/bash


for i in *.2.txt
do


echo $i;
cat $i | ./prune.pl $1 > ${i/.2.txt/_prune$1.txt}
done
#rm -f LPAC*
```

E.1.4   LPAC #3

```bash
#!/bin/bash


#mount /cdrom
#cp /cdrom/* .


for i in *.1.gz
do
echo $i;
gunzip -c $i > tempfile;
grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 | prep > LPAC3_fullprep${i/.gz/.txt}
cat LPAC3_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 1 2 > LPAC3${i/.gz/.txt}
rm -f $i
done


rm -f tempfile
#rm -f LPAC*


#umount /cdrom
#eject
```

E.1.5 FOSP

```
#!/bin/bash


for i in *.gz
do
echo $i;
gunzip -c $i > tempfile;
grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 | prep > FO_fullprep${i/.gz/.txt}
cat FO_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > FO_Temp${i/.gz/.txt}
grep -v ^k FO_Temp${i/.gz/.txt} > FO${i/.gz/.txt}
rm -f $i
rm -f FO_Temp${i/.gz/.txt}
done
rm -f tempfile
```


E.1.6 LOSP

```
#!/bin/bash


for i in *.gz
do
echo $i;
gunzip -c $i > tempfile;
  grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 > LO_fullprep${i/.gz/.txt}
cat LO_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > LO_Temp${i/.gz/.txt}
grep -v ^k LO_Temp${i/.gz/.txt} > LO${i/.gz/.txt}
rm -f $i
rm -f LO_Temp${i/.gz/.txt}
done
#rm -f tempfile;
```

E.1.7 Panel Aggregate

```bash
#!/bin/bash

#mount /cdrom

#cp /cdrom/*.gz .
for i in *.1.gz
do
if [ -e ${i/.1.gz/.2.gz} ] && [ -e ${i/.1.gz/.3.gz} ]
then

echo $i;
gunzip -c $i > tempfile1;
echo ${i/.1.gz/.2.gz};
gunzip -c ${i/.1.gz/.2.gz} > tempfile2;
echo ${i/.1.gz/.3.gz};
gunzip -c ${i/.1.gz/.3.gz} > tempfile3;
grep -v ^# tempfile1 > tempfile1a;
rm -f tempfile1;
grep -v ^# tempfile2 > tempfile2a;
rm -f tempfile2;
grep -v ^# tempfile3 > tempfile3a;
rm -f tempfile3;
paste tempfile1a tempfile2a tempfile3a > tempfile;
cat tempfile | ./chansift2_12bit.pl 1 2 | prep  > UEC2A_fullprep${i/.gz/.txt}
cat tempfile | ./chansift2_12bit.pl 1 3 | prep  > UEC2B_fullprep${i/.gz/.txt}
cat tempfile | ./chansift2_12bit.pl 1 4 | prep  > SCU_fullprep${i/.gz/.txt}
cat tempfile | ./chansift2_12bit.pl 1 5 | prep  > Panel_fullprep${i/.gz/.txt}
cat UEC2A_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > UEC2A${i/.gz/.txt}
cat UEC2B_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > UEC2B${i/.gz/.txt}
cat SCU_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > SCU${i/.gz/.txt}
```

```
cat Panel_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > Panel_${i/.gz/.txt}
rm -f tempfile1a;
rm -f tempfile2a;
rm -f tempfile3a;
rm -f tempfile;
rm -f $i
rm -f ${i/.1.gz/.2.gz}
rm -f ${i/.1.gz/.3.gz}

done
#umount /cdrom
#eject
```

E.1.8   GTG

```
#!/bin/bash

for i in *.gz
do
echo $i;
gunzip -c $i > tempfile;
  grep -v ^# tempfile | ./chansift2_12bit.pl 1 2 | prep > GTG_fullprep${i/.gz/.txt}
cat GTG_fullprep${i/.gz/.txt} | ./chansift2_12bit.pl 2 > GTG${i/.gz/.txt}
rm -f $i
done
rm -f tempfile
```

E.1.9   *chansift2₁2bit.pl*

This script selects columns of data in the file as specified by the user.

```
#!/usr/bin/perl -w

while (<STDIN>) {
```

```
    @cols = split;
    foreach $k (@ARGV) {
#$var = int(($cols[$k-1]));
        $var = ($cols[$k-1]);
print "$var \t";
    }
    print "\n";
}
```

E.2   Matlab Scripts to Plot Data

E.2.1   Example of Plotting Data for all Files for 1SA NILM

```
% A test script to run through an entire directory

D=dir('*.txt');
Ptot = [];
for i=1:1:length(D)

    V = 440;
    Rm = 110;
    K = 1000;
    g = 5;

    P = [V*prep_to_current(g, Rm, K, load(D(i).name))];
    t = [0:1:length(P)-1]*(1/120)*(1/60);

    plot(t,P);
    saveas(gcf, regexprep(D(i).name,'txt','fig'));
    Ptot=[Ptot;P];
end
Ttot = [0:1:length(Ptot)-1]*(1/120)*(1/60);
plot(Ttot,Ptot);
```

```
saveas(gcf, regexprep('1SATotalPower.fig'));
```

E.2.2 Example of Time Aligned Plot of Multiple Source Files

```
%~~~~~~~~~~~~~~~
% Plot 3SA Data
%~~~~~~~~~~~~~~~

    x = load ('3SAsnapshot-20080505-140001.1.txt');
    V = 440;
    Rm = 130;
    K = 1000;
    g = 1.25;


    P = [V*prep_to_current(g, Rm, K, x)];
    t = [0:1:length(P)-1]*(1/120)*(1/60);


    h(1) = subplot(5,1,1);
    plot(t,-P);
    title('UEC2A');
    xlabel('Time (min)');
    ylabel('Power (W)');


%~~~~~~~~~~~~~~~~~~~
% Plot LOSP 2A Data
%~~~~~~~~~~~~~~~~~~~

    x = load ('LOsnapshot-20080505-140001.1.txt');
    V = 440;
    Rm = 20;
    K = 2500;
    g = 10;


    P = [V*prep_to_current(g, Rm, K, x)];
    t = [0:1:length(P)-1]*(1/120)*(1/60);
```

```
    h(2) = subplot(5,1,2);

    plot(t,-P);

    title('LOSP 2A');

    xlabel('Time (min)');

    ylabel('Power (W)');


%~~~~~~~~~~~~~~~~~~
% Plot FOSP 2A Data
%~~~~~~~~~~~~~~~~~~

    x = load ('FOsnapshot-20080505-140001.1.txt');

    V = 440;

    Rm = 61;

    K = 1000;

    g = 10;


    P = [V*prep_to_current(g, Rm, K, -x)];

    t = [0:1:length(P)-1]*(1/120)*(1/60);


    h(3) = subplot(5,1,3);

    plot(t,P);

    title('FOSP 2A');

    xlabel('Time (min)');

    ylabel('Power (W)');


%~~~~~~~~~~~~~~~~~~~~
% Plot #2 LPAC Data
%~~~~~~~~~~~~~~~~~~~~

    x = load ('LPAC2snapshot-20080505-140001.1.txt');

    V = 440;

    Rm = 36;

    K = 2500;
```

```matlab
    g = 10;

    P = [V*prep_to_current(g, Rm, K, -x)];
    t = [0:1:length(P)-1]*(1/120)*(1/60);

    h(4) = subplot(5,1,4);
    plot(t,P);
    title('LPAC #2');
    xlabel('Time (min)');
    ylabel('Power (W)');


%~~~~~~~~~~~~~~~~~~
% Plot #3 LPAC Data
%~~~~~~~~~~~~~~~~~~
    x = load ('LPAC3snapshot-20080505-140001.1.txt');
    V = 440;
    Rm = 13.5;
    K = 2500;
    g = 10;

    P = [V*prep_to_current(g, Rm, K, -x)];
    t = [0:1:length(P)-1]*(1/120)*(1/60);

    h(5) = subplot(5,1,5);
    plot(t,P);
    title('LPAC #3');
    xlabel('Time (min)');
    ylabel('Power (W)');

saveas(gcf, '3SATimeAlignPlot','fig');
```

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix F

CHT GUI Matlab Code

F.1 Main NILM GUI Program

```
function varargout = NILM_GUI(varargin)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                         %
%                    INITIALIZATION TASKS OF THE GUI                      %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
      GUI_LAYOUT
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Create the figure, the indications and the system diagram
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
DrawSystemDiagramGUI()
InitializeVariables()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Find and stop all running timer functions to prevent multiples running
% on the system should it be shutdown incorrectly
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
timers=timerfind;
delete(timerfind);


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Initialize the timer function
% - the timer will execute functions automatically to keep the data
%   up to date
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
t_main = timer('period',2.0,'ExecutionMode','fixedrate','StartDelay',0.5);
```

```
t_main.TimerFcn = ({@MyTimerFunction});

start(t_main);


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Initialize the alarm timer function

% - the timer will execute functions automatically to keep the data

%   up to date

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

t_alarm = timer('period',4.0,'ExecutionMode','fixedrate','StartDelay',0.5);

t_alarm.TimerFcn = (@AlarmSound);

start(t_alarm);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%                                                                            %

%                      UTILITY FUNCTIONS OF THE GUI                          %

%                                                                            %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Main GUI Function

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function MainGUI()


%~~~~~~~~~~~~~~~

% Update the GUI

%~~~~~~~~~~~~~~~

UpdateGUIData()


%~~~~~~~~~~~~~~~~~~~~~~~~

% Check for New *.evt Files

%~~~~~~~~~~~~~~~~~~~~~~~~
```

113

```
CheckForNewFiles()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% End Main GUI Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% Play the Alarm Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
function AlarmSound(obj, event)
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        GUI_LAYOUT


if (ALARMS.SHOW_WARNING)

    sound(PARAMETERS.ALARM_SOUND);

end
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% End Play the Alarm Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
% Update GUI Data Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%
function UpdateGUIData()
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        GUI_LAYOUT


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    %
    % Show correct pump states for running conditions
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    if (STATE.VP.ON)

        set(PICTURES.RUNNING.VP.PIC,'Visible', 'on');
```

```matlab
else
    set(PICTURES.RUNNING.VP.PIC,'Visible', 'off');
end


if (STATE.DP.ON)
    set(PICTURES.RUNNING.DP.PIC,'Visible', 'on');
else
    set(PICTURES.RUNNING.DP.PIC,'Visible', 'off');
end


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Show correct states for alarm conditions
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if (ALARMS.PF.ACTIVE)
    set(PICTURES.ALARMS.PF.PIC, 'Visible', 'on');
    set(GUI_LAYOUT.Warning, 'String', ALARMS.PF.WARNING, ...
        'BackgroundColor', 'red');
    set(GUI_LAYOUT.ResetButton, 'Visible', 'on');
    ALARMS.SHOW_WARNING = true;
end
if (ALARMS.CSW.ACTIVE)
    set(PICTURES.ALARMS.CSW.PIC, 'Visible', 'on');
    set(GUI_LAYOUT.Warning, 'String', ALARMS.CSW.WARNING, ...
        'BackgroundColor', 'red');
    set(GUI_LAYOUT.ResetButton, 'Visible', 'on');
    ALARMS.SHOW_WARNING = true;
end
if (ALARMS.CGL.ACTIVE)
    set(PICTURES.ALARMS.CGL.PIC, 'Visible', 'on');
    set(GUI_LAYOUT.Warning, 'String', ALARMS.CGL.WARNING, ...
        'BackgroundColor', 'red');
    set(GUI_LAYOUT.ResetButton, 'Visible', 'on');
```

```matlab
        ALARMS.SHOW_WARNING = true;
end
if (ALARMS.FL.ACTIVE)
    set(PICTURES.ALARMS.FL.PIC, 'Visible', 'on');
    set(GUI_LAYOUT.Warning, 'String', ALARMS.FL.WARNING, ...
        'BackgroundColor', 'red');
    set(GUI_LAYOUT.ResetButton, 'Visible', 'on');
    ALARMS.SHOW_WARNING = true;
end
if (ALARMS.SL.ACTIVE)
    set(PICTURES.ALARMS.SL.PIC, 'Visible', 'on');
    set(GUI_LAYOUT.Warning, 'String', ALARMS.SL.WARNING, ...
        'BackgroundColor', 'red');
    set(GUI_LAYOUT.ResetButton, 'Visible', 'on');
    ALARMS.SHOW_WARNING = true;
end
if (~ALARMS.SHOW_WARNING)
    set(GUI_LAYOUT.ResetButton, 'Visible', 'off');
end


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Display the information for the current event
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[m,n] = size(EVENTS_LIST);

set(GUI_LAYOUT.DateValue,'String', datestr(EVENTS_LIST(n).DATE_TIME,1));
set(GUI_LAYOUT.TimeValue,'String', datestr(EVENTS_LIST(n).DATE_TIME,13));
set(GUI_LAYOUT.EventValue,'String', EVENTS_LIST(n).EVENT_TEXT);


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Update the statistical information for the pumps
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```matlab
if (isfield(EVENT_STATS.VP, 'RUN_TIME'))

    ART_VP = num2str(round(mean([EVENT_STATS.VP.RUN_TIME])));

    set(GUI_LAYOUT.VPARTValue, 'String', strcat(ART_VP, ' sec'));

end


if (isfield(EVENT_STATS.DP, 'RUN_TIME'))

    ART_DP = num2str(round(mean([EVENT_STATS.DP.RUN_TIME])));

    set(GUI_LAYOUT.DPARTValue, 'String', strcat(ART_DP, ' sec'));

end


Time24hrs = EVENTS_LIST(n).DATE_TIME - 1;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Delete all times in the TRACKING.VP.RUNS24 structure which are
% > 24hrs old
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if (max(TRACKING.VP.RUNS24) > 0)

    OldTimeMax = 1;
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Find the highest index of the time > 24hrs ago
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    while (TRACKING.VP.RUNS24(OldTimeMax)< Time24hrs)

            TRACKING.VP.RUNS24(OldTimeMax) < Time24hrs;

            OldTimeMax = OldTimeMax + 1;

    end


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Delete the times > 24hrs ago
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    if (OldTimeMax > 1)

        TRACKING.VP.RUNS24(1:OldTimeMax)=[];

    end
```

```
        [m,n] = size(TRACKING.VP.RUNS24);

        set(GUI_LAYOUT.VP24HValue, 'String', n);

    end


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    % Delete all times in the TRACKING.DP.RUNS24 structure which are

    % > 24hrs old

    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    %

    if (max(TRACKING.DP.RUNS24) > 0)
%          for i=1:length(TRACKING.DP.RUNS24)
%              if (EVENTS_LIST(n).DATE_TIME(i) < Time24hrs)
%                  TRACKING.DP.RUNS24(i) = [];
%              end
%          end


        [m,n] = size(TRACKING.DP.RUNS24);

        set(GUI_LAYOUT.DP24HValue, 'String', n);

    end


    drawnow


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% End Update GUI Data Function

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% My Timer Function

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

function MyTimerFunction(obj, event)

global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        GUI_LAYOUT
```

```
MainGUI()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% End My Timer Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%   Function To Check For New Data Files
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function CheckForNewFiles()

    global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
            GUI_LAYOUT


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Check for new files in directory
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    FileList = dirc(strcat(PARAMETERS.NEWDATA,'/*.evt'),'','d');
    FileListSize = size(FileList);


    while (FileListSize(1) > 0)
        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        % If there are data files in the directory, read the information
        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        LongFileName = strcat(PARAMETERS.NEWDATA, char(FileList(1,1)));
        [DateTimeStamp, EventCode, event_class, window_size, P, Q, local_index] = ...
                                        Read_File_Info(LongFileName);


        DateNum = datenum(DateTimeStamp, 'yyyymmdd-HH:MM:SS');
        DateVector = datevec(DateNum);


        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        %
```

```matlab
% Append data to Event Data Matrices
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[m,n] = size(EVENTS_LIST);

EVENTS_LIST(n+1).DATE_TIME = DateNum;

EVENTS_LIST(n+1).TIMESTAMP = DateTimeStamp;

EVENTS_LIST(n+1).EVENTCODE = str2num(EventCode);

EVENTS_LIST(n+1).FILENAME = char(FileList(1,1));

EVENTS_LIST(n+1).EVENT_TEXT = event_class;


%~~~~~~~~~~~~~~~~~~~~
% Classify the Event
%~~~~~~~~~~~~~~~~~~~~
EventClassification(n+1)


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Call the Diagnostic Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Diagnostics(n+1)


%~~~~~~~~~~~~~~~~
% Update the GUI
%~~~~~~~~~~~~~~~~
UpdateGUIData()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% move files to Old Data directory
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
FN_from = strcat(PARAMETERS.NEWDATA, char(FileList(1,1)));

FN_to = strcat(PARAMETERS.OLDDATA, char(FileList(1,1)));

movefile(FN_from, FN_to);
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Delete First Row of FileList Array
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
FileList(1,:)=[];


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Recheck Directory for New Files
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
FileList = dirc('NewData/*.evt','','d');
FileListSize = size(FileList);
end


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% End of Check For New Files Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                         %
%                    CLOSING FUNCTIONS OF THE GUI                         %
%                                                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End My Functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Non-Intrusive Load Monitor (NILM)
%    Initialize Variables Function
%    Written by: Richard Jones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function InitializeVariables()


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
       TEMPRUN


%~~~~~~~~~~~
% PARAMETERS
%~~~~~~~~~~~
PARAMETERS.OLDDATA = 'OldData/';

PARAMETERS.NEWDATA = 'NewData/';

PARAMETERS.PICTURES = 'Pictures/';

PARAMETERS.BACKUP = 'DataBackup/';

PARAMETERS.SOUNDS = 'Sounds/';

PARAMETERS.SYSTEM_PICTURE = 'CHT_SYSTEM_DP_FAULT.jpg';

PARAMETERS.ALARM_SOUND = wavread('Sounds/DangerNew.wav');

PARAMETERS.INITIALIZE = false;


%~~~~~~
% STATE
%~~~~~~
STATE.VP.ON = 0;

STATE.DP.ON = 0;

STATE.VP.START = false;

STATE.VP.STOP = false;
```

```
STATE.DP.START = false;

STATE.DP.STOP = false;

STATE.VP.ERROR =  false;

STATE.DP.ERROR = false;


%~~~~~~~
% ALARMS
%~~~~~~~
ALARMS.CGL.ACTIVE = false;

ALARMS.CGL.WARNING = 'Clogged Gauge Line Detected';

ALARMS.CGL.VISIBLE = false;

ALARMS.PF.ACTIVE = false;

ALARMS.PF.WARNING = 'Probe Failure Detected';

ALARMS.PF.VISIBLE = false;

ALARMS.PF.LIST = 0;

ALARMS.SL.ACTIVE = false;

ALARMS.SL.WARNING = 'Slow Leak Detected in the System';

ALARMS.SL.VISIBLE = false;

ALARMS.FL.ACTIVE = false;

ALARMS.FL.WARNING = 'Fast Leak Detected in the System';

ALARMS.FL.VISIBLE = false;

ALARMS.CSW.ACTIVE = false;

ALARMS.CSW.WARNING = 'Clogged Seal Water Line to the Vacuum Pumps Detected';

ALARMS.CSW.VISIBLE = false;

ALARMS.SHOW_WARNING = false;


%~~~~~~~~~
% TRACKING
%~~~~~~~~~
TRACKING.VP.RUNS24 = 0;

TRACKING.DP.RUNS24 = 0;

TRACKING.VP.AVERUNTIME = 0;
```

```
    TRACKING.DP.AVERUNTIME = 0;


%~~~~~~~~~~~~
% EVENTS LIST
%~~~~~~~~~~~~
[m,n] = size(EVENTS_LIST);
if (n < 2)
    EVENTS_LIST.DATE_TIME = 0;
    EVENTS_LIST.TIMESTAMP = '';
    EVENTS_LIST.EVENTCODE = 0;
    EVENTS_LIST.FILENAME = '';
    EVENTS_LIST.EVENT_TEXT = '';
end


%~~~~~~~~~~~~
% EVENT STATS
%~~~~~~~~~~~~
if (isfield(EVENT_STATS, 'VP'))
    [m,n] = size(EVENT_STATS.VP);
else
    n = 0;
end

if (n < 2)
    EVENT_STATS.VP.START = 0;
    EVENT_STATS.VP.STOP = 0;
    EVENT_STATS.VP.START_FILENAME = '';
    EVENT_STATS.VP.STOP_FILENAME = '';
    EVENT_STATS.VP.RUN_TIME = 0;
    EVENT_STATS.VP.OFF_TIME = 0;
end
```

```
if (isfield(EVENT_STATS, 'DP'))
    [m,n] = size(EVENT_STATS.DP);
else
    n = 0;
end


if (n < 2)
    EVENT_STATS.DP.START = 0;
    EVENT_STATS.DP.STOP = 0;
    EVENT_STATS.DP.START_FILENAME = '';
    EVENT_STATS.DP.STOP_FILENAME = '';
    EVENT_STATS.DP.RUN_TIME = 0;
    EVENT_STATS.DP.OFF_TIME = 0;
end


%~~~~~~~~~
% PICTURES
%~~~~~~~~~
PICTURES.ALARMS.CGL.SHOW = false;
PICTURES.ALARMS.PF.SHOW = false;
PICTURES.ALARMS.CSW.SHOW = false;
PICTURES.RUNNING.VP.SHOW = false;
PICTURES.RUNNING.DP.SHOW = false;
for i=1:3
    PICTURES.ALARMS.SL(i).SHOW = false;
end
for i=1:3
    PICTURES.ALARMS.FL(i).SHOW = false;
end


%~~~~~~~~~
% TEMPRUN
```

```
%~~~~~~~~

if (~isfield(TEMPRUN, 'VP'))

    TEMPRUN.VP.TIME = 0;

    TEMPRUN.VP.FILE = '';

end

if (~isfield(TEMPRUN, 'DP'))

    TEMPRUN.DP.TIME = 0;

    TEMPRUN.DP.FILE = '';

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% End of Function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
function DrawSystemDiagramGUI()
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT


scrsz = get(0,'ScreenSize');
GUIWidth = 750;
GUIHeight = 600;
GUI_LAYOUT.MainFigure = figure(...        % The main GUI figure
                        'MenuBar','none', ...
                        'Toolbar','none', ...
                        'HandleVisibility','callback', ...
                        'Color', get(0,'defaultuicontrolbackgroundcolor'), ...
                        'Position', [(scrsz(3)-GUIWidth)/2 (scrsz(4)-GUIHeight)/2 GUIWidth
                        'Name','Non-Intrusive Load Monitor (NILM)', ...
                        'NumberTitle', 'off', ...
                        'HandleVisibility','on');
GUI_LAYOUT.PlotAxes = axes(...     % Axes for plotting the system diagram
                        'Parent', GUI_LAYOUT.MainFigure, ...
                        'Units', 'normalized', ...
                        'HandleVisibility','callback', ...
                        'Position',[0.0 0.01 0.85 0.90], ...
                        'DataAspectRatioMode', 'manual', ...
                        'PlotBoxAspectRatio', [1 1 1], ...
                        'PlotBoxAspectRatioMode', 'manual', ...
                        'XTick', [], 'YTick', [], ...
                        'NextPlot', 'add', ...
                        'HandleVisibility','on');
GUI_LAYOUT.ResetButton = uicontrol(GUI_LAYOUT.MainFigure,'Style', 'pushbutton', ...
                        'String', 'Reset', ...
                        'Position', [660 480 90 30], ...
                        'Callback', {@AlarmReset_Callback}, ...
                        'FontSize', 12, 'FontWeight', 'bold', ...
```

```matlab
                               'Visible', 'off');

GUI_LAYOUT.CloseButton = uicontrol(GUI_LAYOUT.MainFigure,'Style', 'pushbutton', ...
                               'String', 'Quit NILM', ...
                               'Position', [660 30 90 30], ...
                               'Callback', {@ExitButton_Callback}, ...
                               'FontSize', 12, 'FontWeight', 'bold', ...
                               'Value', 1);

GUI_LAYOUT.VPARTTitle = uicontrol( ...   % Text Title for VP Ave Run Time
                               GUI_LAYOUT.MainFigure, ...
                               'Style', 'text', ...
                               'String', 'VP Ave Run Time', ...
                               'Units', 'Normalized', ...
                               'Position', [0.80, 0.65, 0.2, 0.04], ...
                               'BackgroundColor', 'blue', ...
                               'ForegroundColor', 'yellow', ...
                               'FontSize', 12, 'FontWeight', 'bold');

GUI_LAYOUT.VPARTValue = uicontrol( ...   % Text Title for VP Ave Run Time
                               GUI_LAYOUT.MainFigure, ...
                               'Style', 'text', ...
                               'String', 'VP ART', ...
                               'Units', 'Normalized', ...
                               'Position', [0.80, 0.61, 0.2, 0.04], ...
                               'BackgroundColor', 'white', ...
                               'FontSize', 12);

GUI_LAYOUT.VP24HTitle = uicontrol( ...   % Text Title for VP Ave Run Time
                               GUI_LAYOUT.MainFigure, ...
                               'Style', 'text', ...
                               'String', 'last 24hr VP Runs', ...
                               'Units', 'Normalized', ...
                               'Position', [0.80, 0.55, 0.2, 0.04], ...
                               'BackgroundColor', 'blue', ...
                               'ForegroundColor', 'yellow', ...
```

```matlab
                           'FontSize', 12, 'FontWeight', 'bold');
GUI_LAYOUT.VP24HValue = uicontrol( ...   % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'VP 24h', ...
                           'Units', 'Normalized', ...
                           'Position', [0.80, 0.51, 0.2, 0.04], ...
                           'BackgroundColor', 'white', ...
                           'FontSize', 12);
GUI_LAYOUT.DPARTTitle = uicontrol( ...   % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'DP Ave Run Time', ...
                           'Units', 'Normalized', ...
                           'Position', [0.80, 0.35, 0.2, 0.04], ...
                           'BackgroundColor', 'blue', ...
                           'ForegroundColor', 'yellow', ...
                           'FontSize', 12, 'FontWeight', 'bold');
GUI_LAYOUT.DPARTValue = uicontrol( ...   % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'DP ART', ...
                           'Units', 'Normalized', ...
                           'BackgroundColor', 'white', ...
                           'Position', [0.80, 0.31, 0.2, 0.04], ...
                           'FontSize', 12);
GUI_LAYOUT.DP24HTitle = uicontrol( ...   % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'last 24hr DP Runs', ...
                           'Units', 'Normalized', ...
                           'Position', [0.80, 0.25, 0.2, 0.04], ...
```

```matlab
                              'BackgroundColor', 'blue', ...
                              'ForegroundColor', 'yellow', ...
                              'FontSize', 12, 'FontWeight', 'bold');
    GUI_LAYOUT.DP24HValue = uicontrol( ...   % Text Title for VP Ave Run Time
                              GUI_LAYOUT.MainFigure, ...
                              'Style', 'text', ...
                              'String', 'DP 24h', ...
                              'Units', 'Normalized', ...
                              'Position', [0.80, 0.21, 0.2, 0.04], ...
                              'BackgroundColor', 'white', ...
                              'FontSize', 12);
    GUI_LAYOUT.DateTitle = uicontrol( ...   % Text Title for VP Ave Run Time
                              GUI_LAYOUT.MainFigure, ...
                              'Style', 'text', ...
                              'String', 'Date:', ...
                              'Units', 'Normalized', ...
                              'Position', [0.00, 0.0, 0.10, 0.04], ...
                              'BackgroundColor', 'blue', ...
                              'ForegroundColor', 'yellow', ...
                              'FontSize', 12, 'FontWeight', 'bold');
    GUI_LAYOUT.DateValue = uicontrol( ...   % Text Title for VP Ave Run Time
                              GUI_LAYOUT.MainFigure, ...
                              'Style', 'text', ...
                              'String', 'Date Here', ...
                              'Units', 'Normalized', ...
                              'Position', [0.10, 0.0, 0.15, 0.04], ...
                              'BackgroundColor', 'white', ...
                              'FontSize', 12);
    GUI_LAYOUT.TimeTitle = uicontrol( ...   % Text Title for VP Ave Run Time
                              GUI_LAYOUT.MainFigure, ...
                              'Style', 'text', ...
                              'String', 'Time:', ...
```

```matlab
                           'Units', 'Normalized', ...
                           'Position', [0.25, 0.0, 0.10, 0.04], ...
                           'BackgroundColor', 'blue', ...
                           'ForegroundColor', 'yellow', ...
                           'FontSize', 12, 'FontWeight', 'bold');
   GUI_LAYOUT.TimeValue = uicontrol( ...  % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'Time Here', ...
                           'Units', 'Normalized', ...
                           'Position', [0.35, 0.0, 0.15, 0.04], ...
                           'BackgroundColor', 'white', ...
                           'FontSize', 12);
   GUI_LAYOUT.EventTitle = uicontrol( ...  % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'Event:', ...
                           'Units', 'Normalized', ...
                           'Position', [0.50, 0.0, 0.10, 0.04], ...
                           'BackgroundColor', 'blue', ...
                           'ForegroundColor', 'yellow', ...
                           'FontSize', 12, 'FontWeight', 'bold');
   GUI_LAYOUT.EventValue = uicontrol( ...  % Text Title for VP Ave Run Time
                           GUI_LAYOUT.MainFigure, ...
                           'Style', 'text', ...
                           'String', 'Event Here', ...
                           'Units', 'Normalized', ...
                           'Position', [0.60, 0.0, 0.45, 0.04], ...
                           'BackgroundColor', 'white', ...
                           'FontSize', 12);
   GUI_LAYOUT.Warning = uicontrol( ...  % Warning text
                           GUI_LAYOUT.MainFigure, ...
```

```matlab
                              'Style', 'text', ...

                              'String', '', ...

                              'Units', 'Normalized', ...

                              'Position', [0.0, 0.85, 1.0, 0.05], ...

                              'BackgroundColor', 'blue', ...

                              'ForegroundColor', 'white', ...

                              'FontSize', 16, 'FontWeight', 'bold');

GUI_LAYOUT.NILMTitle = uicontrol( ...  % NILM Title text

                              GUI_LAYOUT.MainFigure, ...

                              'Style', 'text', ...

                              'String', 'Non-Intrusive Load Monitoring', ...

                              'Units', 'Normalized', ...

                              'Position', [0.0, 0.90, 1.0, 0.10], ...

                              'BackgroundColor', 'blue', ...

                              'ForegroundColor', 'white', ...

                              'FontSize', 18, 'FontWeight', 'bold');


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Create object groups for controlling group settings in the GUI

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

GUI_LAYOUT.AlarmGroup = hggroup;

GUI_LAYOUT.RunningGroup = hggroup;


hold(GUI_LAYOUT.PlotAxes, 'on');

axis(GUI_LAYOUT.PlotAxes, [0, 800, 0, 600]);

axis(GUI_LAYOUT.PlotAxes, 'off');


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```matlab
% Update the System Diagram
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


%~~~~~~~~~~~~
% Draw Border
%~~~~~~~~~~~~
%rectangle('Position',[0,0, 800, 600]);


%~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Draw all Circular objects
%~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Vacuum Pump
AA(1).x = 635;
AA(1).y = 400;
AA(1).r = 50;
AA(1).color = 'red';


% Discharge Pump
AA(2).x = 635;
AA(2).y = 200;
AA(2).r = 50;
AA(2).color = 'green';


% Gauge Body
AA(3).x = 50;
AA(3).y = 400;
AA(3).r = 30;
AA(3).color = 'green';


[m,n] = size(AA);


for i = 1:n
```

```
        OpenCircleGUI(AA(i).x, AA(i).y, AA(i).r);

        drawnow

end


%~~~~~~~~~~~~~~~~~

% Alarm Conditions

%~~~~~~~~~~~~~~~~~

% Vacuum Pump

AA(1).x = 635;

AA(1).y = 400;

AA(1).r = 50;

AA(1).color = 'red';

PICTURES.ALARMS.CSW.PIC = FillCircleGUI(AA(1).x, AA(1).y, AA(1).r, AA(1).color);

set(PICTURES.ALARMS.CSW.PIC, 'Visible', 'off', 'Parent', GUI_LAYOUT.AlarmGroup);

drawnow


% Discharge Pump

AA(2).x = 635;

AA(2).y = 200;

AA(2).r = 50;

AA(2).color = 'yellow';

%PICTURES.ALARMS.CSW.PIC = FillCircleGUI(AA(2).x, AA(2).y, AA(2).r, AA(2).color);

%set(PICTURES.ALARMS.CSW.PIC, 'Visible', 'off', 'Parent', GUI_LAYOUT.AlarmGroup);

drawnow


% Gauge Body

AA(3).x = 50;

AA(3).y = 400;

AA(3).r = 30;

AA(3).color = 'red';

PICTURES.ALARMS.CGL.PIC = FillCircleGUI(AA(3).x, AA(3).y, AA(3).r, AA(3).color);

set(PICTURES.ALARMS.CGL.PIC, 'Visible', 'off', 'Parent', GUI_LAYOUT.AlarmGroup);
```

```
drawnow


%~~~~~~~~~~~~~~~~~~~
% Running Conditions
%~~~~~~~~~~~~~~~~~~~~~
% Vacuum Pump
AA(1).x = 635;
AA(1).y = 400;
AA(1).r = 50;
AA(1).color = 'green';
PICTURES.RUNNING.VP.PIC = FillCircleGUI(AA(1).x, AA(1).y, AA(1).r, AA(1).color);
set(PICTURES.RUNNING.VP.PIC, 'Visible', 'off', ...
                             'Parent', GUI_LAYOUT.RunningGroup);
drawnow


% Discharge Pump
AA(2).x = 635;
AA(2).y = 200;
AA(2).r = 50;
AA(2).color = 'green';
PICTURES.RUNNING.DP.PIC = FillCircleGUI(AA(2).x, AA(2).y, AA(2).r, AA(2).color);
set(PICTURES.RUNNING.DP.PIC, 'Visible', 'off', ...
                             'Parent', GUI_LAYOUT.RunningGroup);
drawnow


%~~~~~~~~~~~~~~~~
% Vacuum Receiver
%~~~~~~~~~~~~~~~~
% Normal Vacuum Receiver
rectangle('Position',[100,100,300,400],'Curvature',[.3], 'LineWidth', 4);
drawnow
```

```
% Alarm Condition for Vacuum Receiver
PICTURES.ALARMS.PF.PIC = rectangle('Position',[100,100,300,400],'Curvature', ...
                  [.3], 'LineWidth', 4, 'FaceColor','r');
set(PICTURES.ALARMS.PF.PIC, 'Visible', 'off', 'Parent', GUI_LAYOUT.AlarmGroup, ...
                  'ButtonDownFcn', @PF_ButtonDownFcn);
drawnow


%~~~~~~
% Gauge
%~~~~~~
% Gauge Line
X = [80, 100];
Y = [400, 400];
line(X, Y, 'Color', 'black', 'LineWidth', 8);


% Gauge Needle
X = [35, 65];
Y = [415, 385];
line(X, Y, 'Color', 'black', 'LineWidth', 2);


X = [35, 35];
Y = [405, 415];
line(X, Y, 'Color', 'black', 'LineWidth', 2);


X = [35, 45];
Y = [415, 415];
line(X, Y, 'Color', 'black', 'LineWidth', 2);


%~~~~~~~~~~~~~~
% Level Probes
%~~~~~~~~~~~~~~
X = [275, 275];
```

```matlab
Y = [150, 500];
line(X, Y, 'Color', 'blue', 'LineWidth', 3);


X = [250, 250];
Y = [250, 500];
line(X, Y, 'Color', 'blue', 'LineWidth', 3);


X = [225, 225];
Y = [350, 500];
line(X, Y, 'Color', 'blue', 'LineWidth', 3);


%~~~~~~~~~~
% Fast Leak
%~~~~~~~~~~
X_FL(1).x1 = 100;
X_FL(1).x2 = 75;
Y_FL(1).y1 = 550;
Y_FL(1).y2 = 575;


X_FL(2).x1 = 100;
X_FL(2).x2 = 100;
Y_FL(2).y1 = 550;
Y_FL(2).y2 = 585;


X_FL(3).x1 = 100;
X_FL(3).x2 = 125;
Y_FL(3).y1 = 550;
Y_FL(3).y2 = 575;


for i=1:3
    PICTURES.ALARMS.FL(i).PIC = line([X_FL(i).x1, X_FL(i).x2], ...
                                      [Y_FL(i).y1, Y_FL(i).y2], ...
```

```matlab
                                        'Color', 'red', 'LineWidth', 4);
    set(PICTURES.ALARMS.FL(i).PIC, 'Visible', 'off', ...
                                  'Parent', GUI_LAYOUT.AlarmGroup);
    drawnow
end


%~~~~~~~~~~
% Slow Leak
%~~~~~~~~~~
X_SL(1).x1 = 150;
X_SL(1).x2 = 135;
Y_SL(1).y1 = 550;
Y_SL(1).y2 = 565;


X_SL(2).x1 = 150;
X_SL(2).x2 = 150;
Y_SL(2).y1 = 550;
Y_SL(2).y2 = 575;


X_SL(3).x1 = 150;
X_SL(3).x2 = 165;
Y_SL(3).y1 = 550;
Y_SL(3).y2 = 565;


for i=1:3
    PICTURES.ALARMS.SL(i).PIC = line([X_SL(i).x1, X_SL(i).x2], ...
                                     [Y_SL(i).y1, Y_SL(i).y2], ...
                                     'Color', 'red', 'LineWidth', 4);
    set(PICTURES.ALARMS.SL(i).PIC, 'Visible', 'off', ...
                                  'Parent', GUI_LAYOUT.AlarmGroup);
end
```

```
%~~~~~~~~~~~~~~~~~~~~~~~
% Draw in Piping System
%~~~~~~~~~~~~~~~~~~~~~~~
% Vacuum Piping
X = [50, 550];
Y = [550, 550];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [250, 250];
Y = [500, 550];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [550, 550];
Y = [550, 400];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [550, 635];
Y = [400, 400];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [630, 700];
Y = [450, 450];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


% Discharge Piping
X = [250, 550];
Y = [50, 50];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [250, 250];
Y = [50, 100];
line(X, Y, 'Color', 'black', 'LineWidth', 4);
```

139

```matlab
X = [550, 550];
Y = [50, 200];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [550, 635];
Y = [200, 200];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


X = [630, 700];
Y = [250, 250];
line(X, Y, 'Color', 'black', 'LineWidth', 4);


function h = FillCircleGUI(x,y,r, color)
    global GUI_LAYOUT
    hold on
    th = 0:pi/50:2*pi;
    xunit = r * cos(th) + x;
    yunit = r * sin(th) + y;
    h = fill(xunit, yunit, color, ...
        'LineWidth', 4, ...
        'Parent', GUI_LAYOUT.PlotAxes);
    hold off


function h = OpenCircleGUI(x,y,r)
    global GUI_LAYOUT
    hold on
    th = 0:pi/50:2*pi;
    xunit = r * cos(th) + x;
    yunit = r * sin(th) + y;
    h = plot(xunit, yunit,'Color', 'black', 'LineWidth', 4, 'Parent', GUI_LAYOUT.PlotAxes);
    hold off
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Alarm Reset Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function AlarmReset_Callback(hObject, eventdata)
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT ALARM_NOW

    %~~~~~~~~~~~~~~~~~~~~~~~~~
    % Clear All Active Alarms
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.PF.ACTIVE = false;
    ALARMS.CSW.ACTIVE = false;
    ALARMS.CGL.ACTIVE = false;
    ALARMS.FL.ACTIVE = false;
    ALARMS.SL.ACTIVE = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Set All Alarm Pictures to 'off'
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    set(PICTURES.ALARMS.PF.PIC, 'Visible', 'off');
    set(PICTURES.ALARMS.CSW.PIC, 'Visible', 'off');
    set(PICTURES.ALARMS.CGL.PIC, 'Visible', 'off');

    for i=1:3
        set(PICTURES.ALARMS.FL(i).PIC, 'Visible', 'off');
        set(PICTURES.ALARMS.SL(i).PIC, 'Visible', 'off');
    end


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Reset the Alarm Banner back to original blue w/o text
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
        set(GUI_LAYOUT.Warning, 'String', '', ...
                          'BackgroundColor', 'blue');


    %~~~~~~~~~~~~~~~~~~~~
    % Turn of alarm sound
    %~~~~~~~~~~~~~~~~~~~~
    ALARMS.SHOW_WARNING = false;


    %~~~~~~~~~~~~~~~~~
    % Hide Reset Button
    %~~~~~~~~~~~~~~~~~
    set(GUI_LAYOUT.ResetButton, 'Visible', 'off');


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Exit Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function ExitButton_Callback(hObject, eventdata)
% hObject    handle to ExitButton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Backup all data prior to closing
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% BackupData


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Find and stop all running timer functions
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
timers=timerfind;
delete(timerfind);
close all;
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Probe Failure Button Down Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function PF_ButtonDownFcn(hObject, eventdata)
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT ALARM_NOW
    %~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Clear PF Active Alarms
    %~~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.PF.ACTIVE = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Set PF Alarm Pictures to 'off'
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%     set(PICTURES.ALARMS.PF.PIC, 'Visible', 'off');


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Reset the Alarm Banner back to original blue w/o text
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    set(GUI_LAYOUT.Warning, 'String', '', ...
                            'BackgroundColor', 'blue');


    %~~~~~~~~~~~~~~~~~~~~~~~~
    % Turn of alarm sound
    %~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.SHOW_WARNING = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Open the Alarm Information GUI
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.ALARMCODE = 'PF';
```

```
    DrawAlarmGUI()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Clogged Gauge Line Button Down Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function CGL_ButtonDownFcn(hObject, eventdata)
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT ALARM_NOW
    %~~~~~~~~~~~~~~~~~~~~~~~~
    % Clear CGL Active Alarms
    %~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.CGL.ACTIVE = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Set PF Alarm Pictures to 'off'
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%     set(PICTURES.ALARMS.CGL.PIC, 'Visible', 'off');


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Reset the Alarm Banner back to original blue w/o text
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    set(GUI_LAYOUT.Warning, 'String', '', ...
                            'BackgroundColor', 'blue');


    %~~~~~~~~~~~~~~~~~~~~~~
    % Turn of alarm sound
    %~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.SHOW_WARNING = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Open the Alarm Information GUI
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```matlab
    ALARMS.ALARMCODE = 'CGL';

    DrawAlarmGUI()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Clogged Seal Water Button Down Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function CSW_ButtonDownFcn(hObject, eventdata)
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT ALARM_NOW
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    % Clear CSW Active Alarms
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.CSW.ACTIVE = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Set CSW Alarm Pictures to 'off'
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%   set(PICTURES.ALARMS.CSW.PIC, 'Visible', 'off');


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Reset the Alarm Banner back to original blue w/o text
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    set(GUI_LAYOUT.Warning, 'String', '', ...
                            'BackgroundColor', 'blue');


    %~~~~~~~~~~~~~~~~~~~~~~
    % Turn of alarm sound
    %~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.SHOW_WARNING = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Open the Alarm Information GUI
```

```matlab
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.ALARMCODE = 'CSW';
    DrawAlarmGUI()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Fast Leak Button Down Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function FL_ButtonDownFcn(hObject, eventdata)
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT ALARM_NOW
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    % Clear FL Active Alarms
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.FL.ACTIVE = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Set FL Alarm Pictures to 'off'
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%    set(PICTURES.ALARMS.FL.PIC, 'Visible', 'off');


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Reset the Alarm Banner back to original blue w/o text
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    set(GUI_LAYOUT.Warning, 'String', '', ...
                            'BackgroundColor', 'blue');


    %~~~~~~~~~~~~~~~~~~~~~~~
    % Turn of alarm sound
    %~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.SHOW_WARNING = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
    % Open the Alarm Information GUI
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.ALARMCODE = 'FL';
    DrawAlarmGUI()


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Slow Leak Button Down Function
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
function SL_ButtonDownFcn(hObject, eventdata)
global  PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        VP_ON VPON GUI_LAYOUT ALARM_NOW
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    % Clear SL Active Alarms
    %~~~~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.SL.ACTIVE = false;


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Set SL Alarm Pictures to 'off'
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%    set(PICTURES.ALARMS.SL.PIC, 'Visible', 'off');


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Reset the Alarm Banner back to original blue w/o text
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    set(GUI_LAYOUT.Warning, 'String', '', ...
                            'BackgroundColor', 'blue');


    %~~~~~~~~~~~~~~~~~~~~~~
    % Turn of alarm sound
    %~~~~~~~~~~~~~~~~~~~~~~
    ALARMS.SHOW_WARNING = false;
```

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Open the Alarm Information GUI
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ALARMS.ALARMCODE = 'SL';
DrawAlarmGUI()
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Non-Intrusive Load Monitor (NILM)
%   Event Classification Function
%
%   Written by: Richard Jones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function EventClassification(EventNum)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Local Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
STATE.VP.ERROR = false;
STATE.DP.ERROR = false;


%~~~~~~~~~~~~~~~~~~~~~~
% Event Code Functions
%~~~~~~~~~~~~~~~~~~~~~~
switch (EVENTS_LIST(EventNum).EVENTCODE)
    case 0      % Unclassifiable
    case 1      % VP On
        STATE.VP.ON = STATE.VP.ON + 1;
        STATE.VP.START = true;
        PICTURES.RUNNING.VP.SHOW = true;
    case 2      % VP Off
        STATE.VP.STOP = true;
        PICTURES.RUNNING.VP.SHOW = false;
    case 3      % DP On
```

```
        STATE.DP.ON = STATE.DP.ON + 1;

        STATE.DP.START = true;

        PICTURES.RUNNING.DP.SHOW = true;

case 4       % DP Off

        STATE.DP.STOP = true;

        PICTURES.RUNNING.DP.SHOW = false;

case 5       % VP On (Clogged)

        STATE.VP.ON = STATE.VP.ON + 1;

        STATE.VP.START = true;

        PICTURES.RUNNING.VP.SHOW = true;

        STATE.VP.ERROR = true;

case 7       % Unclassifiable On Event

case 9       % All Off Event

        STATE.VP.STOP = true;

        STATE.DP.STOP = true;

        PICTURES.RUNNING.VP.SHOW = false;

        PICTURES.RUNNING.DP.SHOW = false;

case 11      % 2 VPs On

        STATE.VP.ON = STATE.VP.ON + 2;

        STATE.VP.START = true;

        PICTURES.RUNNING.VP.SHOW = true;

case 13      % VP on and DP on

        STATE.VP.ON = STATE.VP.ON + 1;

        STATE.VP.START = true;

        STATE.DP.ON = STATE.DP.ON + 1;

        STATE.DP.START = true;

        PICTURES.RUNNING.VP.SHOW = true;

        PICTURES.RUNNING.DP.SHOW = true;

case 21      % VP On (Normal) followed by a it immediately turning off

        STATE.VP.ON = STATE.VP.ON + 1;

        STATE.VP.START = true;

        STATE.VP.STOP = true;
```

```matlab
            STATE.VP.ERROR = true;
    case 43      % DP On followed by it immediately turning off
        STATE.DP.ON = STATE.DP.ON + 1;

        STATE.DP.START = true;

        STATE.DP.STOP = true;

        STATE.DP.ERROR = true;
    case 65      % VP On (Clogged) followed by it turning immediately off
        STATE.VP.ON = STATE.VP.ON + 1;

        STATE.VP.START = true;

        STATE.VP.STOP = true;

        STATE.VP.ERROR = true;
    case 87      % Unclassifiable On Event followed by an Off Event
    otherwise    % The Catch-All in the case of unknown code
end


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Add the Date/Time stamp to the RUNS24 structure if a pump started
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
[m,n] = size(TRACKING.VP.RUNS24);

if (STATE.VP.START)

    if (n == 1) && (TRACKING.VP.RUNS24(n) == 0)

        TRACKING.VP.RUNS24(n) = EVENTS_LIST(EventNum).DATE_TIME;

    else

        TRACKING.VP.RUNS24(n+1) = EVENTS_LIST(EventNum).DATE_TIME;

    end

end


[m,n] = size(TRACKING.DP.RUNS24);

if (STATE.DP.START)

    if (n == 1) && (TRACKING.DP.RUNS24(n) == 0)

        TRACKING.DP.RUNS24(n) = EVENTS_LIST(EventNum).DATE_TIME;

    else
```

151

```
            TRACKING.DP.RUNS24(n+1) = EVENTS_LIST(EventNum).DATE_TIME;
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of Function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Non-Intrusive Load Monitor (NILM)
%    Function To Perform Leak Analysis of the CHT System
%
%    Written by: Richard Jones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Diagnostics(n)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
        TEMPRUN


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Initialize process state variables
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ContinueProcessDP = true;
ContinueProcessVP = true;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagnostic Logic
% If the CHT system is determined to not be operating correctly, the leak
% detection should not be performed.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (STATE.VP.ERROR)
    ContinueProcessVP = false;
end


if (STATE.DP.ERROR)
    ContinueProcessDP = false;
end
```

```matlab
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Update the tracking arrays as applicable
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if (STATE.VP.START) && (ContinueProcessVP)
    TempRun(n);
end


if (STATE.DP.START) && (ContinueProcessDP)
    TempRun(n);
end


if (STATE.VP.STOP) && (ContinueProcessVP)
    ConstructVPArray;
end


if (STATE.DP.STOP) && (ContinueProcessDP)
    ConstructDPArray;
end




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Probe Failure Detection
% If there is a failure of the level sensing probes the in the Vacuum tank,
% the Discharge Pumps will cycle on and off frequently, only running for a
% short period of time (spikes in power data).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (EVENTS_LIST(n).EVENTCODE == 43)
    ALARMS.PF.ACTIVE = true;
    [M_num, num] = size(ALARMS.PF.LIST);
    if (num == 1) && (ALARMS.PF.LIST(num) == 0)
        ALARMS.PF.LIST(num) = n;
```

```
        else

            ALARMS.PF.LIST(num + 1) = n;

        end

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Clogged Gauge Line Detection
% If the gauge line becomes clogged, the Vacuum Pumps will run for an
% extended time to lower the tank vacuum to a level which  will reach the
% setpoint of the control circuit for Vacuum Pump operation.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Update the Gauge Clog Array if VP RunTime > 2min (120 Sec)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
EVENT_STATS.VP(n).RUN_TIME;
if (EVENT_STATS.VP(n).RUN_TIME > 120)

    if (isfield(GAUGE_CLOG, 'Array'))

        [m,n] = size(GAUGE_CLOG.Array);


        if (n == 5)

            for i = 1:4

                GAUGE_CLOG.Array(n) = GAUGE_CLOG.Array(n+1);

            end

        end

        if n == 0

            n = 1;

        end

    else

        n = 1;

    end


    GAUGE_CLOG(1).Array(n) = EVENT_STATS(n).VP_Stop;
```

```
end


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Check for 5 flags set in the Array,
% and check time of oldest entry if
% 5 flags are present
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if (isfield(GAUGE_CLOG, 'Array'))
    if (size(GAUGE_CLOG.Array) == 5)


disp('I''m here :)');


        if (GAUGE_CLOG.Array(1) < 3600)
            GAUGE_CLOG.Warning(1).Show = 1;
            GAUGE_CLOG.Warning(1).Text = 'Clogged Gauge Line Detected';
            ProcessLeak = 0;
        else
            GAUGE_CLOG.Warning(1).Show = 0;
            GAUGE_CLOG.Warning(1).Text = '';
        end

    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Leak Detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ProcessLeak = 0;
%if (isfield(PUMP_RUNS, 'VP_OFF'))
%    if (PUMP_RUNS.VP_OFF == 1)
%        if (ProcessLeak)
%            LeakAnalysis(n);
%        end
```

```
%    end
%end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of Function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Non-Intrusive Load Monitor (NILM)
%    Function to construct the array for the DischargePumps (DP).  This is
%    then used for diagnostic purposes.
%
%    Written by: Richard Jones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ConstructDPArray()


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
       TEMPRUN


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Only Perform the calulations if there is a VP running,
% otherwise, ignore the VP shutting off
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if (STATE.DP.ON > 0)
    %~~~~~~~~~~~~~~~~~~~~~
    % Find size of arrays
    %~~~~~~~~~~~~~~~~~~~~~
        [m,n] = size(EVENTS_LIST);
        [m,NumEvents] = size(EVENT_STATS.DP);


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Update the DP Statistics When the DP is Secured
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        EVENT_STATS.DP(NumEvents+1).START = TEMPRUN(1).DP.TIME;
        EVENT_STATS.DP(NumEvents+1).STOP = EVENTS_LIST(n).DATE_TIME;
        EVENT_STATS.DP(NumEvents+1).START_FILENAME = TEMPRUN.DP(1).FILE;
```

158

```
                EVENT_STATS.DP(NumEvents+1).STOP_FILENAME = EVENTS_LIST(n).FILENAME;


        %~~~~~~~~~~~~~~~~~~~~~~~~~
        % Compute the Pump Runtime
        %~~~~~~~~~~~~~~~~~~~~~~~~~
            Delta = datevec(EVENT_STATS.DP(NumEvents+1).STOP - ...
                            EVENT_STATS.DP(NumEvents+1).START);
            EVENT_STATS.DP(NumEvents+1).RUN_TIME = Delta(5) * 60 + Delta(6);


        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        % Compute the time between DP runs
        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            if (NumEvents > 1)
                Delta = datevec(EVENT_STATS.DP(NumEvents+1).START - ...
                                EVENT_STATS.DP(NumEvents).STOP);

                EVENT_STATS.DP(NumEvents+1).OFF.TIME = Delta(4)/60 + Delta(5) + Delta(6)*60;
            else
                EVENT_STATS.DP(NumEvents+1).OFF.TIME = 0;
            end


        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        % Reset the tracking variables
        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            STATE.DP.ON = 0;
            STATE.DP.STOP = false;

else
    STATE.DP.STOP = false;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of Function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Non-Intrusive Load Monitor (NILM)
%   Function to construct the array for the Vacuum Pumps (VP).  This is
%   then used for diagnostic purposes.
%
%   Written by: Richard Jones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ConstructVPArray()


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global PARAMETERS STATE ALARMS TRACKING EVENTS_LIST EVENT_STATS PICTURES ...
       TEMPRUN


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Only Perform the calulations if there is a VP running,
% otherwise, ignore the VP shutting off
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
if (STATE.VP.ON > 0)
    %~~~~~~~~~~~~~~~~~~~~~
    % Find size of arrays
    %~~~~~~~~~~~~~~~~~~~~~
    [m,n] = size(EVENTS_LIST);
    [m,NumEvents] = size(EVENT_STATS.VP);


    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    % Update the VP Statistics When the VP is Secured
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    %~~~~~~~~~
    % Pump #1
    %~~~~~~~~~
```

161

```
        EVENT_STATS.VP(NumEvents+1).START = TEMPRUN.VP(1).TIME;

        EVENT_STATS.VP(NumEvents+1).STOP = EVENTS_LIST(n).DATE_TIME;

        EVENT_STATS.VP(NumEvents+1).START_FILENAME = TEMPRUN.VP(1).FILE;

        EVENT_STATS.VP(NumEvents+1).STOP_FILENAME = EVENTS_LIST(n).FILENAME;


%~~~~~~~~~~~~~~~~~~~~~~~~~~
% Compute the Pump Runtime
%~~~~~~~~~~~~~~~~~~~~~~~~~~

        Delta = datevec(EVENT_STATS.VP(NumEvents+1).STOP - ...
                        EVENT_STATS.VP(NumEvents+1).START);

        EVENT_STATS.VP(NumEvents+1).RUN_TIME = Delta(5) * 60 + Delta(6);


%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Compute the time between VP runs
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        if (NumEvents > 1)
            Delta = datevec(EVENT_STATS.VP(NumEvents+1).START - ...
                            EVENT_STATS.VP(NumEvents).STOP);


            EVENT_STATS.VP(NumEvents+1).OFF_TIME = Delta(4)/60 + Delta(5) + Delta(6)*60;
        else
            EVENT_STATS.VP(NumEvents+1).OFF_TIME = 0;
        end


%~~~~~~~~~
% Pump #2
%~~~~~~~~~

    if (STATE.VP.ON == 2)
        EVENT_STATS.VP(NumEvents+2).START = TEMPRUN.VP(2).TIME;
        EVENT_STATS.VP(NumEvents+2).STOP = EVENTS_LIST(n).DATE_TIME;
        EVENT_STATS.VP(NumEvents+2).START_FILENAME = TEMPRUN.VP(2).FILE;
        EVENT_STATS.VP(NumEvents+2).STOP_FILENAME = EVENTS_LIST(n).FILENAME;
```

162

```matlab
        %~~~~~~~~~~~~~~~~~~~~~~~~~~
        % Compute the Pump Runtime
        %~~~~~~~~~~~~~~~~~~~~~~~~~~

            Delta = datevec(EVENT_STATS.VP(NumEvents+2).STOP - ...
                            EVENT_STATS.VP(NumEvents+1).START);
            EVENT_STATS.VP(NumEvents+2).RUN_TIME = Delta(5) * 60 + Delta(6);


        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        % Compute the time between VP runs
        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

            Delta = datevec(EVENT_STATS.VP(NumEvents+2).START - ...
                            EVENT_STATS.VP(NumEvents).STOP);


            EVENT_STATS.VP(NumEvents+2).OFF_TIME = Delta(4)/60 + Delta(5) + Delta(6)*60;
        end


        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        % Reset the tracking variables
        %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
            STATE.VP.ON = 0;
            STATE.VP.STOP = false;


else
    STATE.VP.STOP = false;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of Function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    Non-Intrusive Load Monitor (NILM)
%    Function To Backup All Data
%
%    Written by: Richard Jones
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function BackupData()


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global EVENTS_LIST EVENT_STATS EVENTS_SIZE EVENTLIST EVENTTIMES VPRUN ...
       DPRUN PUMP_RUNS SHOW_WARNING TOTAL_PUMP_RUNS SLOW_LEAK


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Save EVENT_STATS data structure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save DataBackup/BackupEventStats.mat EVENT_STATS;
save DataBackup/TotalPumpRuns.mat TOTAL_PUMP_RUNS;
save DataBackup/SlowLeak.mat SLOW_LEAK;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% End of Function
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```